NASA CONTRACTOR REPORT

NASA CR

NASA CR-1798

TECHNIQUES FOR GENERATION OF CONTROL AND GUIDANCE SIGNALS DERIVED FROM OPTICAL FIELDS

Part II

by H. Hemami, R. B. McGhee, and S. R. Gardner

Prepared by
THE OHIO STATE UNIVERSITY
RESEARCH FOUNDATION
Columbus, Ohio 43210
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • NOVEMBER 1971

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-1798		

4. Title and Subtitle	5. Report Date
TECHNIQUES FOR GENERATION OF CONTROL AND GUIDANCE SIGNALS DERIVED FROM OPTICAL FIELDS - PART II	November 1971
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
H. Hemami, R. B. McGhee, S. R. Gardner	Final Report RF 2421
	10. Work Unit No.
9. Performing Organization Name and Address	
The Ohio State University Research Foundation 1659 N High Columbus, OH 43210	11. Contract or Grant No. NGR 36-008-076
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report is concerned with the development of a high resolution technique for the detection and identification of landmarks from spacecraft optical fields. By making use of nonlinear regression analysis, a method is presented whereby a sequence of synthetic images produced by a digital computer can be automatically adjusted to provide a least squares approximation to a real image. The convergence of the method is demonstrated by means of a computer simulation for both elliptical and rectangular patterns.

Statistical simulation studies with elliptical and rectangular patterns show that the computational techniques developed are able to at least match human pattern recognition capabilities, even in the presence of large amounts of noise. Unlike most pattern recognition techniques, this ability is unaffected by arbitrary pattern rotation, translation, and scale change. Further development of the basic approach established by this research should eventually allow a spacecraft or robot vehicle to be provided with an ability to very accurately determine its spatial relationship to arbitrary known objects within its optical field of view.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Pattern recognition **Control and guidance information** **Optical fields**	**Unclassified - Unlimited**

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	**Unclassified**	142	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

FOREWORD

The study described in this report was conducted at the Communications and Control Systems Laboratory of The Ohio State University, and was supported by a grant from the National Aeronautics and Space Administration under Grant No. NGR-36-008-076. The NASA scientific officer was Mr. Thomas W. Walsh.

The study was performed under the supervision of H. Hemami and R. B. McGhee. Numerous fruitful discussions with Professor K. J. Breeding are gratefully acknowledged.

TABLE OF CONTENTS

FIGURES

TABLES

CHAPTER I

INTRODUCTION

The problem of acquisition and identification of a landmark within a given field of view is treated here from two points of view:

1. Identification of a landmark:

2. Estimation of its translation and rotation with respect to the reference frame.

One application of this approach is to a navigation problem. One may have a photograph of an island and the coordinates and altitude at which the photograph was recorded. If at a later time a camera carrying vehicle flies over this island at the same altitude, then as the island comes into the field of view of the camera, one can, by the approach presented here, estimate position (translation) and orientation (rotation) of the craft with respect to the island. These estimates then could be used to command the propulsion system and navigate the vehicle.

In addition to applications in landmark identification and acquisition,this approach is potentially useful in problems of automatic docking since it permits measurement of rotation and translation of the docking target with respect to docking craft (command module). This means a television camera onboard the docking craft takes a picture of the docking target. By detection of the rotation and translation of the docking target with respect to the stored reference, the docking craft can position itself for automatic docking.

The approach presented here requires edge enhancement so that the boundaries of the landmark are detected and uses the information contained in the boundary of the pattern by successively reading the coordinates of the boundary and developing a nonlinear regression analysis technique for simultaneous estimation of rotation and translation of the landmark. This method appears to be very sensitive and offers high resolution both in rotation and translation parameters.

In this research only two-dimensional landmarks or patterns were considered. Specifically patterns in the form of ellipses and rectangles were first considered. The motivation for these two classes was to consider a class of simple shapes that can be analytically represented, and another class that could not be analytically represented. In addition, different amounts of sensor noise and measurement noise were added to the coordinates of the boundary points to check the performance of this method under a variety of circumstances.

While some alternative methods such as detection of centroid, etc., may be more useful for recognition of rectangles and ellipses, noting that by recognition we mean measuring rotation, translation and size of a pattern, it was felt necessary to consider a more general approach that would be applicable to more classes of two dimensional patterns as well as three-dimensional patterns.

Since the method is based on tracking the boundary of a pattern, a review of the state of the art in boundary tracking and its application in pattern recognition, estimation, etc., is provided in Chapter II under the title of quantization and encoding of arbitrary curves. In Chapter III, the fundamental nonlinear regression analysis approach is discussed. In Chapter IV the recognition of elliptical planar patterns is presented. In Chapter V the recognition of rectangular planar patterns is discussed. Conclusions, a summary of the results, and potential future research areas are discussed in Chapter VI. For use of interested readers, the main computer program is also documented in an appendix.

CHAPTER II

QUANTIZATION AND ENCODING OF ARBITRARY CURVES

This chapter is concerned with a review of approaches to boundary tracking and implementation of the boundary information in recognition, coding, estimation, etc. A more common name associated with this area has been contour tracing which has been used in the field of pattern recognition and specifically in feature extraction techniques.

In the general problem of pattern recognition, many researchers feel that "contours carry a significant fraction of the information required for recognition of image objects"[1] . Since the recognition scheme developed in this research uses contour information exclusively, it seems appropriate to review some of the work which has been done in this area.

Since most pattern recognition schemes are carried out on a digital computer, it is necessary to be able to represent a pattern in a form which may be easily manipulated by a digital computer. More specifically, if one is given a pictorial representation of some planar configuration, it is desirable to quantize and encode the boundary curve of this pattern into a form such that the digital computer can easily find such properties of the pattern as area, length of the boundary curve, width, height, and others to be discussed later.

A great deal of work in quantizing and encoding arbitrary plane curves has been done by many researchers. [2 -9] It is the intent of this chapter to review some of this research, particularly that of the chain representation as developed by H. Freeman. [10-16]

Of the many ways in which an arbitrary planar curve (assumed to be continuous) may be quantized, a particularly simple technique is called the grid-intersect quantization method. In this method the curve is placed over a square grid, and the grid node lying closest to the point of intersection of the curve with a given grid line is considered to be a point on the quantized curve. Such a grid node is called a curve point. This procedure is illustrated in Figure 1, where the separation between adjacent grid nodes is T.

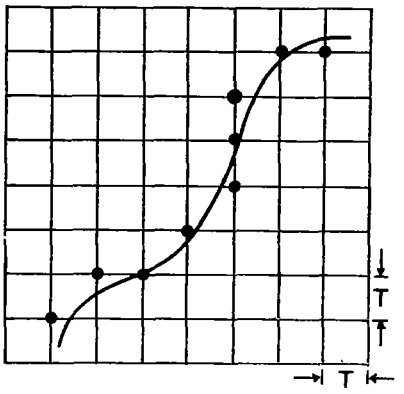


Fig. 1--Node points for a continuous curve.

The lines connecting adjacent curve points have length T or $\sqrt{2}\,T$, as seen from Figure 1. The quantized curve becomes a better approximation to the original curve as the grid separation, T, becomes small compared to the smallest instantaneous radius of curvature of the original curve. Freeman [11] points out that the grid intersect quantization method has an advantage over similar quantization techniques in that it comes the closest to giving equal probability to the occurrence of adjacent curve points which are diagonal. For an arbitrary curve one would expect one half of the adjacent curve points to be connected by diagonal lines and one half to be connected by horizontal and vertical lines.

Once the curve points have been determined, it is desirable to encode these points in some manner that affords economy in computer storage requirements and permits analytical manipulations of the pattern to be accomplished. One obvious encoding would be to simply store the coordinates of each of the curve points. However, even for a relatively coarse grid (say 1024 by 1024), each curve point would require 10 bits for each of its coordinates. A more economical encoding scheme takes advantage of the fact that since the curve which was quantized is continuous, then successive curve points must be adjacent, as shown pictorially in Figure 2. The center node is assumed to be a curve point and the next curve point must be one of the eight nodes shown.

If the straight lines which join the center node with each of the surrounding eight nodes are assigned the same number as the corresponding outer node, then the original curve may be represented by a sequence of short line segments, with each line segment encoded by an integer

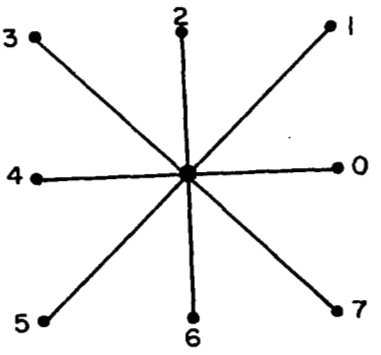


Fig. 2--Numbering scheme for adjacent curve points.

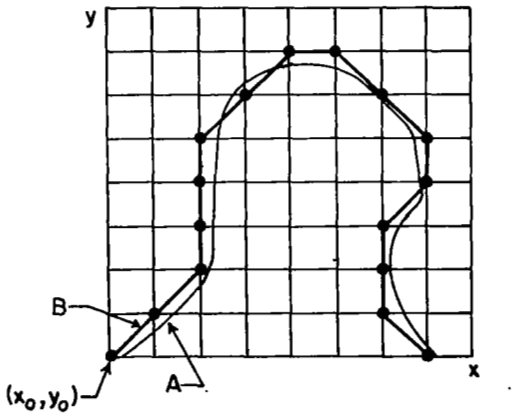


Fig. 3--Chain representation of a continuous curve.

between 0 and 7. A line segment connecting two adjacent curve points is referred to as an element, and the sequence of elements which represents the curve is called a chain. Thus, curve A in Figure 3 may be represented by the straight line segment curve B which is characterized by the chain 112221107765667. If the absolute location of curve A is required with respect to the x-y coordinate system, then the starting point, or initium, of the chain must be specified. In this case it is denoted by (x_0, y_0). It is apparent that the dimension of the measurement space has been drastically reduced since now, with the exception of the starting point, each element (and therefore each curve point) requires only 3 bits of computer storage to specify it compared to 20 bits which are necessary to specify the coordinates of each curve point. Actually, if the value of a given element is known, then the next element, in general, will not assume each of its eight possible values with equal probability. This fact can be used to further increase the coding efficiency by employing a chain-difference encoding scheme. [11]

The chain B, which is the straight line segment representation of curve A in Figure 3, may be written using the chaining or "concatenation" operator C defined by

$$B = b_1 b_2 - - - b_n = \overset{n}{\underset{i=1}{C}} b_i$$

where $b_i = 0, 1, 2, 3, 4, 5, 6,$ or 7

and the element b_i connects curve points i-1 and i. It is apparent that the number of elements in a chain will be proportional to the length of the curve and inversely proportional to the grid separation, T. Furthermore, for a curve that is quantized into n curve points, the associated chain will have n-1 elements if the curve is open, and n elements if the curve is closed. It is also readily apparent that the angle which an element makes with respect to the positive x-axis is simply the element value multiplied by 45°.

Before considering some of the properties which chains possess, a few of the ambiguities in the chain representation should be pointed out. Consider, for instance, the chain given in Figure 3. If the absolute position of this chain is not important, then the coordinates of the initium, (x_0, y_0), can be disregarded and the chain is given by 112221107765667 as before. However, the chain may also be written as 322123345566655. This chain represents exactly the same straight line segment curve, but traced in the reverse direction. Chains possessing this property are called inverses. Similarly, elements whose slopes differ by 180° are called inverse elements. Therefore, the elements

a_i and a_i^{-1} are inverses if

$$a_i^{-1} = a_i \dot{+} 4$$

where the symbol $\dot{+}$ designates modulo eight addition. Thus, an inverse chain is obtained by finding the inverse elements of the elements of the original chain and then reversing the order of the inverse elements.

From the above discussion it is apparent that any simple open curve (no self-intersections) has two chain representations, each being the inverse of the other. A simple closed curve, on the other hand, may be represented by any one of 2n different chains. This is due to the fact that there is no unique starting point for the chain; in fact, the chain can start at any one of the n curve points to give a total of n different chain representations. Some of the ambiguity in the chain representation of a closed curve can be eliminated if, for instance, the curve is always traversed in the clockwise direction, and the starting point is always chosen to be the curve point which is nearest to the origin.

It is now appropriate to consider some of the properties which chains possess. These properties may then be incorporated into a variety of pattern classification schemes. It is seen that a chain is invariant with translation; that is to say, a chain becomes fixed in a coordinate system only after the coordinates of its initium have been specified. A chain may be rotated by $k \cdot 45°$ by the modulo eight addition of k to each element of the chain, where k is an integer. However, the rotation is distortion-free only when k is an even integer, since when k is an odd integer the length of any element in the original chain is changed from T to $\sqrt{2}$ T or vice versa.

The length of a chain may be directly computed by counting the number of even elements, n_e, and the number of odd elements, n_o. Since an even element has length T, while an odd element has length $\sqrt{2}$ T, the length of the chain is simply

$$L = n_e T + n_o \sqrt{2} T$$

$$= (n_e + n_o \sqrt{2}) T$$

If n, the total number of elements of the chain, is large then the length of the chain may be approximated by

$$L \tilde{\sim} (1 + .414 \rho) n T$$

where ρ is the fraction of the adjacent curve points which are diagonal for the particular quantization method being used. Since $\rho = 0.41$ for

grid-intersect quantization [11], the length of a long chain is approximately

$$L \approx 1.17\, n\, T$$

The height and width of a chain may also be simply computed. The x and y components, a_{x_i} and a_{y_i}, of each of the elements a_i are shown as follows:

a_i	a_{x_i}	a_{y_i}
0	T	0
1	T	T
2	0	T
3	-T	T
4	-T	0
5	-T	-T
6	0	-T
7	T	-T

The height is then found by subtracting the chain's largest negative deviation from the x-axis from the largest positive deviation. Thus, the height is given by

$$H = (H_i)_{max} - (H_i)_{min}$$

where $i = 0, 1, 2, \cdots, n$

$$H_i = \sum_{j=1}^{i} a_{y_i}$$

and $H_0 \equiv 0$

Likewise, the width of a chain is given by

$$W = (W_i)_{max} - (W_i)_{min}$$

where $i = 0, 1, 2, \cdots, n$

$$W_i = \sum_{j=1}^{i} a_{x_i}$$

and $W_0 \equiv 0$

Notice that if $H_n = W_n = 0$, then the chain is closed.

The area enclosed by a simple closed chain may also be simply computed. It can be shown that the area is given by [10, 12]

$$\text{Area} = \sum_{i=1}^{n} a_{x_i} \left(H_{i-1} + \tfrac{1}{2} a_{y_i} \right)$$

The area will be a positive number if the chain is traversed in the clockwise direction, and a negative number if the chain is traversed in the counterclockwise direction.

Many other properties of chains may also be determined which can be employed in a pattern recognition scheme. For instance, it is possible to determine the moments of a chain about specified axes, the location of a chain's centroid, and the axes (if any) about which a chain is symmetric. [12]

Two other useful properties involve correlation functions, i.e., autocorrelation and crosscorrelation. The autocorrelation function of a chain $\overset{n}{\underset{i=1}{C}} a_i$ may be defined as

$$\phi_{aa}(j) = \frac{1}{n} \sum_{i=1}^{n} a_i a_{i+j}$$

for $\quad j = 0, \pm 1, \pm 2, \cdots, \pm n$

The product $a_i a_{i+j}$ is defined to be the cosine of the angle between elements a_i and a_{i+j}. For convenience it is assumed that the chain $\overset{n}{\underset{i=1}{C}} a_i$ is periodic, having a maximum period of length n. Thus

$$a_i = a_{i+kn}$$

for $\quad k = 0, \pm 1, \pm 2, \cdots$

The autocorrelation function is therefore defined for all j, being periodic with maximum period of length n.

The crosscorrelation function of two chains $\overset{n}{\underset{i=1}{C}} a_i$ and $\overset{m}{\underset{i=1}{C}} b_i$ may be defined in two ways,

$$\phi_{ab}(j) = \frac{1}{n} \sum_{i=1}^{n} a_i b_i + j$$

and

$$\phi_{ba}(j) = \frac{1}{m} \sum_{i=1}^{m} a_i + j\, b_i$$

depending upon which chain is shifted. Here again, both chains are assumed to be periodic, i.e.,

$$a_i = a_i + kn$$

$$b_i = b_i + km$$

for $k = 0, \pm 1, \pm 2, \cdots$

It is easy to see that the crosscorrelation function is also periodic, having the same period as the length of the chain which is being shifted. Thus,

$$\phi_{ab}(j) = \phi_{ab}(j + km)$$

for $k = 0, \pm 1, \pm 2, \cdots$

If both chains have the same length (n = m), then

$$\phi_{ab}(-j) = \phi_{ba}(j)$$

Since the autocorrelation function is not unique, i.e., several patterns may possess the same autocorrelation function, it may only be used to place the unknown pattern into a class of patterns. The crosscorrelation functions of the unknown pattern and all the patterns within this selected class may then be compared for recognition purposes. Generally the peak of each crosscorrelation function is determined, and recognition is based on the pattern resulting in the maximum peak. The crosscorrelation method has been quite effective in fitting a segment of a curve to a larger curve, provided that the relative scale and orientation are known for both curves. [15]

Another useful property of chains for recognition purposes is the so-called directionality spectrum. This consists of tabulating the number of elements of the chain having values 0 through 7. The directionality spectrum is then found by multiplying the number of odd-valued elements by $\sqrt{2}$, and drawing a bar graph of the results. A normalized directionality spectrum may be obtained by dividing the number of elements having any given value by the total number of elements $(n_e + \sqrt{2}\, n_0)$.

A property of chains which is rotation invariant is the curvature property. [13] The curvature function of the chain $\overset{n}{\underset{i=1}{C}}\, a_i$ is defined

$$u_{i+\frac{1}{2}} = a_{i+1} - a_i + 8k$$

where k is chosen to be -1, 0, or 1 so that

$$\left| u_{i+\frac{1}{2}} \right| \leq 4$$

The sequence $u_{i+\frac{1}{2}}$ is seen to be the slope change (curvature) of the chain $\overset{n}{\underset{i=1}{C}}\, a_i$ as it is traversed.

The number and location of the zero crossings of a smoothed curvature function may then be used for recognition purposes. [13]

CHAPTER III

NONLINEAR REGRESSION ANALYSIS

3.1 Introduction

This chapter is concerned with the basic nonlinear regression method of analysis that was developed for the purpose of landmark tracking. Since the first class of patterns considered are ellipses, in section 3.2 representation of ellipse patterns are discussed. In section 3.3 the parameter estimation problem is formulated, and its characteristics are delineated in sections 3.3.1 through 3.3.6.

3.2 Representation of the Ellipse Pattern

The equation of an ellipse whose major and minor axes are coincident with the w-z coordinate axes, as shown in Fig. 4, is given by

$$g(w,z) = \frac{w^2}{a^2} + \frac{z^2}{b^2} - 1 = 0 \qquad (3.1)$$

or

$$g(w,z) = e_1 w^2 + e_2 z^2 - 1 = 0 \qquad (3.2)$$

where

$$c_1 = \frac{1}{a^2} \quad , \quad e_2 = \frac{1}{b^2} \qquad (3.3)$$

2a = diameter of the ellipse in the w-direction

2b = diameter of the ellipse in the z-direction.

If one wishes to express the equation of this ellipse with respect to an x-y coordinate system as shown in Fig. 5, the following transformation holds

$$w = (x-A)\cos\theta + (y-B)\sin\theta \qquad (3.4)$$

$$z = -(x-A)\sin\theta + (y-B)\cos\theta \qquad (3.5)$$

The equation for the ellipse in the x-y coordinate system then becomes

$$F(x,y) = g(w,z) \;\Big|\; \begin{array}{l} w = (x-A)\cos\theta + (y-B)\sin\theta \\ z = -(x-A)\sin\theta + (y-B)\cos\theta \end{array}$$

$$= e_1[(x-A)\cos\theta + (y-B)\sin\theta]^2 + e_2[-(x-A)\sin\theta + (y-B)\cos\theta]^2 - 1$$

$$= [e_1\cos^2\theta + e_2\sin^2\theta]x^2 + [2(e_1-e_2)\cos\theta\sin\theta]xy$$

$$+ [e_1\sin^2\theta + e_2\cos^2\theta]y^2$$

$$+ [-2e_1\cos\theta(A\cos\theta + B\sin\theta) - 2e_2\sin\theta(A\sin\theta - B\cos\theta)]x$$

$$+ [-2e_1\sin\theta(A\cos\theta + B\sin\theta) + 2e_2\cos\theta(A\sin\theta - B\cos\theta)]y$$

$$+ [e_1(A\cos\theta + B\sin\theta)^2 + e_2(A\sin\theta - B\cos\theta)^2]$$

$$- 1 \qquad (3.6)$$

Eqn. (3.6) contains five parameters which completely describe the ellipse. These parameters are e_1, e_2, A, B, and θ. One notes that Eqn. (3.6) is a nonlinear function of these parameters.

Equation (3.6) may be transformed into a linear function of a new set of parameters via a nonlinear transformation of the parameters. To this end, let the original parameters be denoted by the vector $\vec{c}$ and the new parameters by the vector $\vec{p}$, i.e.,

$$\vec{c} = \begin{bmatrix} e_1 \\ e_2 \\ A \\ B \\ \theta \end{bmatrix} \qquad \vec{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix} \qquad (3.7)$$

and denote the nonlinear transformation by $\vec{\psi}$, i.e.,

$$\vec{c} = \vec{\psi}(\vec{p}) \qquad (3.8)$$

In order to derive the nonlinear transformation of Eqn. (3.8) one may rewrite Eqn. (3.6) as

$$F(x,y) = \rho_1 x^2 + \rho_2 xy + \rho_3 y^2 + \rho_4 x + \rho_5 y + \rho_6 - 1 = 0 \qquad (3.9)$$

where

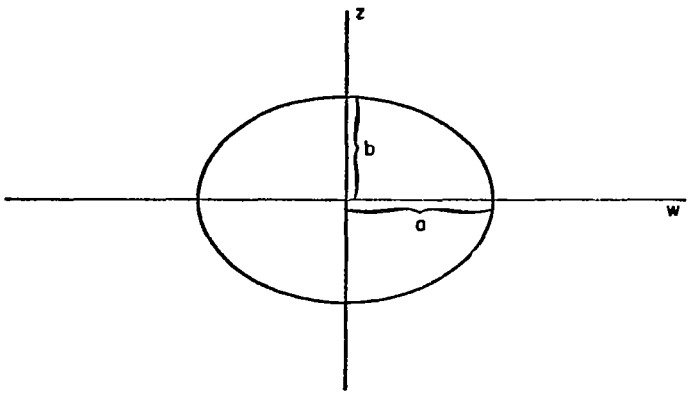


Fig. 4 --Ellipse in reference frame.

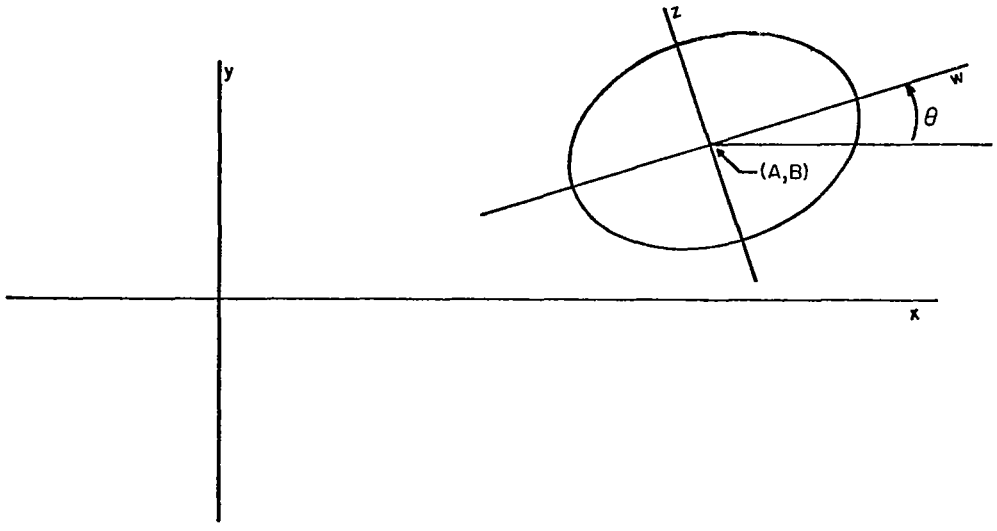


Fig. 5 --Rotated and translated ellipse.

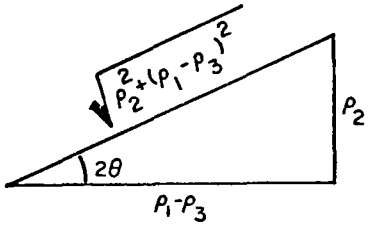


Fig. 6 --Relation of θ and $\vec{\rho}$.

$$\rho_1 = e_1 \cos^2\theta + e_2 \sin^2\theta \tag{3.10}$$

$$\rho_2 = 2(e_1 - e_2) \cos\theta \sin\theta \tag{3.11}$$

$$\rho_3 = e_1 \sin^2\theta + e_2 \cos^2\theta \tag{3.12}$$

$$\rho_4 = -2A(e_1 \cos^2\theta + e_2 \sin^2\theta) - 2B(e_1 - e_2) \cos\theta \sin\theta \tag{3.13}$$

$$\rho_5 = -2B(e_1 \sin^2\theta + e_2 \cos^2\theta) - 2A(e_1 - e_2) \cos\theta \sin\theta \tag{3.14}$$

$$\rho_6 = A^2(e_1 \cos^2\theta + e_2 \sin^2\theta) + B^2(e_1 \sin^2\theta + e_2 \cos^2\theta) + 2AB(e_1 - e_2) \cos\theta \sin\theta \tag{3.15}$$

Eqn. (3.10)-(3.15) may be manipulated to obtain the values of the parameters e_1, e_2, A, B, and θ in terms of the parameters ρ_1, ρ_2, ρ_3, ρ_4, ρ_5, and ρ_6. From Eqn. (3.10), (3.11), and (3.13)

$$\rho_4 = -2A\rho_1 - B\rho_2 \tag{3.16}$$

while Eqn. (3.11), (3.12), and (3.14) yield

$$\rho_5 = -A\rho_2 - 2B\rho_3 \tag{3.17}$$

Solving Eqn. (3.16) and (3.17) simultaneously for A and B gives

$$A = \frac{-2\rho_3\rho_4 + \rho_2\rho_5}{4\rho_1\rho_3 - \rho_2^2} \tag{3.18}$$

$$B = \frac{-2\rho_1\rho_5 + \rho_2\rho_4}{4\rho_1\rho_3 - \rho_2^2} \tag{3.19}$$

From Eqn. (3.10) and (3.12) one obtains

$$\rho_1 + \rho_3 = e_1 + e_2 \tag{3.20}$$

and

$$\rho_1 - \rho_3 = (e_1 - e_2)(\cos^2\theta - \sin^2\theta) = (e_1 - e_2)\cos 2\theta \tag{3.21}$$

while Eqn. (3. 11) yields

$$\rho_2 = (e_1 - e_2) \sin 2\theta \tag{3.22}$$

Eqn. (3. 21) and (3. 22) then give

$$\tan 2\theta = \frac{\rho_2}{\rho_1 - \rho_3} \tag{3.23}$$

or

$$\theta = \tfrac{1}{2} \tan^{-1} \frac{\rho_2}{\rho_1 - \rho_3} \tag{3.24}$$

If a right triangle having sides of length ρ_2 and $\rho_1 - \rho_3$ is formed, then the hypoteneuse has a length $\sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2}$ as shown in Fig. 6. From Fig. 6 it is apparent that

$$\sin 2\theta = \frac{\rho_2}{\sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2}} \tag{3.25}$$

Then from Eqn. (3. 22) and (3. 25) one obtains

$$\sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2} = e_1 - e_2 \tag{3.26}$$

and then Eqn. (3. 20) and (3. 26) yield

$$e_1 = \tfrac{1}{2} (\rho_1 + \rho_3 + \sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2} \tag{3.27}$$

$$e_2 = \tfrac{1}{2} (\rho_1 + \rho_3 - \sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2}) \tag{3.28}$$

Since only five independent parameters are required to fully specify an ellipse, it is reasonable to expect that Eqn. (3. 9) may be simplified. Dividing Eqn. (3. 9) by $(\rho_6 - 1)$ gives

$$F(x, y) = p_1 x^2 + p_2 xy + p_3 y^2 + p_4 x + p_5 y + 1 = 0 \tag{3.29}$$

where

$$p_i = \frac{\rho_i}{\rho_6 - 1} \qquad \text{for } i = 1, 2, 3, 4, 5 \tag{3.30}$$

From Eqn. (3. 18), (3. 19), and (3. 24) one observes that A, B, and θ are ratios of the "ρ" parameters where both numerators and denominators are of the same order. Thus, the denominator of Eqn. (3. 30) will cancel, making Eqn. (3. 18), (3. 19), and (3. 24) the same function of the "p" parameters as of the "ρ" parameters. Therefore

$$A = \frac{-2p_3 p_4 + p_2 p_5}{4 p_1 p_3 - p_2^2} \tag{3.31}$$

$$B = \frac{-2p_1 p_5 + p_2 p_4}{4 p_1 p_3 - p_2^2} \tag{3.32}$$

$$\theta = \frac{1}{2} \tan^{-1} \frac{p_2}{p_1 - p_3} \tag{3.33}$$

This is not the case for Eqn. (3. 27) and (3. 28), however. They become

$$e_1 = (\rho_6 - 1)\left[\frac{1}{2}\left(p_1 + p_3 + \sqrt{p_2^2 + (p_1 - p_3)^2}\right)\right] \tag{3.34}$$

$$e_2 = (\rho_6 - 1)\left[\frac{1}{2}\left(p_1 + p_3 - \sqrt{p_2^2 + (p_1 - p_3)^2}\right)\right] \tag{3.35}$$

From Eqn. (3. 10), (3. 12), and (3. 15)

$$\rho_6 = A^2 \rho_1 + B^2 \rho_3 + AB\rho_2 \tag{3.36}$$

Eqn. (3. 30) and (3. 36) then give

$$\begin{aligned} \rho_6 - 1 &= (A^2 \rho_1 + B^2 \rho_3 + AB\,\rho_2) - 1 \\ &= [(\rho_6 - 1)(A^2 p_1 + B^2 p_3 + AB\,p_2)] - 1 \end{aligned} \tag{3.37}$$

or

$$\rho_6 - 1 = \frac{1}{A^2 p_1 + B^2 p_3 + AB p_2 - 1} \tag{3.38}$$

Making use of Eqn. (3. 31) and (3. 32) further reduces Eqn. (3. 38) to

$$\rho_6 - 1 = \frac{1}{\frac{p_1 p_5^2 + p_3 p_4^2 - p_2 p_4 p_5}{4 p_1 p_3 - p_2^2} - 1}$$

$$= \frac{4 p_1 p_3 - p_2^2}{p_1 p_5^2 + p_3 p_4^2 + p_2^2 - p_2 p_4 p_5 - 4 p_1 p_3} \qquad (3.39)$$

By substituting Eqn. (3.39) into Eqn. (3.34) and (3.35) the parameters e_1 and e_2 become functions only of the "p" parameters.

$$e_1 = \frac{4 p_1 p_3 - p_2^2}{p_1 p_5^2 + p_3 p_4^2 + p_2^2 - p_2 p_4 p_5 - 4 p_1 p_3} \left[\tfrac{1}{2} \left(p_1 + p_3 + \sqrt{p_2^2 + (p_1 - p_3)^2} \right) \right] \qquad (3.40)$$

$$e_2 = \frac{4 p_1 p_3 - p_2^2}{p_1 p_5^2 + p_3 p_4^2 + p_2^2 - p_2 p_4 p_5 - 4 p_1 p_3} \left[\tfrac{1}{2} \left(p_1 + p_3 - \sqrt{p_2^2 + (p_1 - p_3)^2} \right) \right] \qquad (3.41)$$

If the number found from computing Eqn. (3.39) is negative then the expressions for e_1 and e_2 as given by Eqn. (3.40) and 3.41), respectively, should be interchanged as seen by referring to Eqn. (3.21), (3.22), and (3.24).

The derivation of the transformation

$$\vec{c} = \vec{\psi}(\vec{p}) = \begin{bmatrix} \psi_1(\vec{p}) \\ \psi_2(\vec{p}) \\ \psi_3(\vec{p}) \\ \psi_4(\vec{p}) \\ \psi_5(\vec{p}) \end{bmatrix} \qquad (3.42)$$

has now been completed, with the components ψ_1, ψ_2, ψ_3, ψ_4, and ψ_5 given by Eqn. (3.40), (3.41), (3.31), (3.32), and (3.33), respectively.

It was shown that an ellipse may be expressed as a linear function of a set of five parameters or as a nonlinear function of another set of five parameters as given by Eqn. (3.29) and (3.6), respectively. The two sets of parameters are related by the transformation given by Eqn. (3.8). It is now appropriate to investigate methods whereby the unknown parameter vector for an ellipse may be estimated after points on the ellipse have been measured.

3.3 Parameter Estimation Problem

3.3.1 The Error Formulation

In order to estimate the five parameters that represent the size and position of the ellipse one may write Eqn. (3.29) or (3.6) as

$$F(x_i, y_i; \vec{\zeta}_0) = 0 \tag{3.43}$$

where (x_i, y_i) is any point on the ellipse and

$$\vec{\zeta}_0 = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \end{bmatrix} \tag{3.44}$$

is the parameter vector which characterizes the ellipse.

If. Eqn. (3.29) is used for describing the ellipse, then

$$\vec{\zeta}_0 = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix}$$

If one measures any point on this ellipse and computes F using some other parameter vector, $\vec{\zeta}$, the value for F will not be zero, but rather it will be equal to an error, ϵ. That is,

$$F(x_i, y_i ; \vec{\zeta}) = \epsilon \tag{3.45}$$

Likewise, if one makes an error in the measurement of a point on this ellipse, then the computed value for F using the true parameter vector, $\vec{\zeta}_0$, is again non-zero, representing an error, $\mathcal{E}$. That is,

$$F(\tilde{x}_i, \tilde{y}_i ; \vec{\zeta}_0) = \mathcal{E} \tag{3.46}$$

where $(\tilde{x}_i, \tilde{y}_i)$ is now a noisy measurement point.

Now as the boundary of the landmark or the pattern is traced, a sequence of coordinates x_i, y_i become available. Given Eqn. (3.45) and the coordinates of the boundary (x_i, y_i), $i = 1, 2, \cdots$, one can estimate a vector $\vec{\zeta}_e$ of the true parameter vector $\vec{\zeta}_0$.

The estimation will be based on minimizing an appropriate function of the error in Eqn. (3.45). The simplest of these functions appears to be the sum-squared of the error. If N points on the boundary are available, the sum-squared error is given by

$$\phi(\vec{x}, \vec{y} ; \vec{\zeta}) = \sum_{i=1}^{N} [F(\tilde{x}_i, \tilde{y}_i ; \vec{\zeta}) - F(x_i, y_i ; \vec{\zeta}_0)]^2$$

$$= \sum_{i=1}^{N} [F(\tilde{x}_i, \tilde{y}_i ; \vec{\zeta}]^2 \qquad (3.47)$$

For convenience, the tilde on x_i and y_i will be eliminated from now on. It will be understood that (x_i, y_i) represent noisy measurement points. Let

$$\underset{\vec{\zeta}}{\text{Min}}\ \phi(\vec{x}, \vec{y} ; \vec{\zeta}) = \phi(\vec{x}, \vec{y}, \vec{\zeta}_e) \qquad (3.48)$$

where $\vec{x}$, $\vec{y}$ are N-dimensional vectors consisting of the N points which were measured on the ellipse or landmark boundary.

By defining an N x 1 error vector, $\vec{e}$

$$\vec{e} = \vec{F}(\vec{x}, \vec{y} ; \vec{\zeta}) = \begin{bmatrix} F(x_i, y_i ; \vec{\zeta}) \\ \vdots \\ F(x_N, y_N ; \vec{\zeta}) \end{bmatrix} \qquad (3.49)$$

the sum-squared error may be conveniently written as

$$\phi(x, y ; \vec{\zeta}) = \vec{e}^{\,T} \vec{e} \qquad (3.50)$$

where T denotes transposition.

It should be noted that the criterion function, ϕ, corresponding to the sum-squared error is dependent upon whether F given in Eqn. (3.49) corresponds to Eqn. (3.6) or to Eqn. (3.29). The resulting criterion functions are not identical. This point is discussed in greater detail in Appendix II.

3.3.2 Error Minimization by Linear Regression Analysis

When the ellipse is given by Eqn. (3.29) then it is a linear function of the parameter vector, $\vec{p}$, and the minimization of the sum-squared error reduces to a simple result. The error may be written as

$$\vec{e} = \vec{F}(\vec{x}, \vec{y}\,;\, \vec{p}) = M\,\vec{p} + \vec{1} \tag{3.51}$$

where M is the N x 5 matrix

$$M = \begin{bmatrix} x_i^2 & x_i y_i & y_i^2 & x_i & y_i \\ - & - & - & - & - \\ x_N^2 & x_N y_N & y_N^2 & x_N & y_N \end{bmatrix} \tag{3.52}$$

and $\vec{1}$ is an N x 1 vector containing all 1's. Eqn. (3.52) indicates that the elements of M are simply functions of the measured points on the ellipse.

The sum-squared error, ϕ, becomes

$$\begin{aligned} \phi(\vec{x}, \vec{y}\,;\, \vec{p}) &= \vec{e}^{\,T}\,\vec{e} \\ &= (M\,\vec{p} + \vec{1})^T (M\,\vec{p} + \vec{1}) \end{aligned} \tag{3.53}$$

which is a positive definite quadratic form in the coordinates of the trial parameter vector, $\vec{p}$. Therefore, one merely needs to compute all of the partial derivatives of ϕ with respect to the components of $\vec{p}$ and equate them to zero to find the unique minimum value of ϕ. Expanding Eqn. (3.53) gives

$$\begin{aligned} \phi &= (M\,\vec{p})^T (M\,\vec{p}) + (M\,\vec{p})^T\,\vec{1} + \vec{1}^T M\,\vec{p} + \vec{1}^T\,\vec{1} \\ &= \vec{p}^{\,T} M^T M\,\vec{p} + 2\,\vec{p}^{\,T} M^T\,\vec{1} + N \end{aligned} \tag{3.54}$$

and therefore

$$\frac{\partial \phi}{\partial \vec{p}} \triangleq \vec{\nabla}\phi = 2\,M^T M\,\vec{p} + 2\,M^T\,\vec{1} \tag{3.55}$$

and

$$\left.\frac{\partial \phi}{\partial \vec{p}}\right|_{\vec{p} = \vec{p}_e} = 0 = 2\, M^T M \vec{p}_e + 2\, M^T \vec{I} \tag{3.56}$$

or

$$M^T M \vec{p}_e = -M^T \vec{I} \tag{3.57}$$

The matrix $M^T M$ may be inverted, assuming M is of full rank, to give

$$\vec{p}_e = -\left[M^T M\right]^{-1} M^T \vec{I} \tag{3.58}$$

This estimate is then the "least squares estimate" of the true parameter vector, $\vec{p}_0$, and shall be referred to as the one step minimization method.

It should be noted that the simple expression for $\vec{p}_e$ as given in Eqn. (3.58) would not have resulted had the criterion function been something other than quadratic in the parameter vector components, since the differentiation would have yielded a nonlinear relation in the parameter vector components.

If the parameter vector, $\vec{c}_0$, is to be estimated directly from Eqn. (3.6) some other technique than Eqn. (3.58) must be employed since the parameters enter Eqn. (3.6) in a nonlinear manner, making the criterion function, ϕ, no longer quadratic in the parameters. A complete automatic computer algorithm to estimate parameter vectors for this sort of problem has been developed by R. B. McGhee [17] and shall be utilized here. This is an iterative minimization scheme rather than a one step minimization scheme such as was associated with the linear regression analysis. The essence of this computer algorithm is discussed below.

3.3.3 Gauss-Newton Iteration

If the nonlinear response vector, $\vec{F}$, has its Taylor series expansion truncated after the linear term

$$\hat{\vec{F}}(\vec{c}_1 + \Delta \vec{c}) = \vec{F}(\vec{c}_1) + \left.\frac{\partial \vec{F}}{\partial \vec{c}}\right|_{\vec{c} = \vec{c}_1} \Delta \vec{c} \tag{3.59}$$

or

$$\hat{\vec{F}}(\vec{c}_1 + \Delta \vec{c}) = \vec{F}(\vec{c}_1) + Z\, \Delta \vec{c} \tag{3.60}$$

where

$$Z = \left.\frac{\partial \vec{F}}{\partial \vec{c}}\right|_{\vec{c} = \vec{c}_1} = \left[\begin{array}{ccc} \frac{\partial F(x_1, y_1 ; \vec{c})}{\partial c_1} & \cdots & \frac{\partial F(x_1, y_1 ; \vec{c})}{\partial c_5} \\ & & \\ \frac{\partial F(x_N, y_N ; \vec{c})}{\partial c_1} & & \frac{\partial F(x_N, y_N ; \vec{c})}{\partial c_5} \end{array}\right]_{\vec{c} = \vec{c}_1} \tag{3.61}$$

then the criterion function, $\hat{\phi}$, associated with this response function is

$$\hat{\phi}(\vec{x}, \vec{y} ; \Delta \vec{c}, \vec{c}_1) = \hat{\vec{F}}^T \hat{\vec{F}} \tag{3.62}$$

where $\hat{\vec{F}}$ is defined in Eqn. (3.60) and

$$\hat{\phi}(\Delta \vec{c}) = (\vec{e} + Z \Delta \vec{c})^T (\vec{e} + Z \Delta \vec{c}) \tag{3.63}$$

which is a quadratic form in $\Delta \vec{c}$. Expanding Eqn. (3.63) gives

$$\hat{\phi}(\Delta \vec{c}) = \vec{e}^T \vec{e} + \vec{e}^T Z \Delta \vec{c} + \Delta \vec{c}^T Z^T \vec{e} + \Delta \vec{c}^T Z^T Z \Delta \vec{c} \tag{3.64}$$

If Eqn. (3.64) is differentiated with respect to $\Delta \vec{c}$ and the result equated to zero, the minimizing value of $\Delta \vec{c}$ becomes

$$\left.\frac{\partial \hat{\phi}}{\partial \Delta \vec{c}}\right|_{\Delta \vec{c} = \Delta \vec{c}_1} = 0 = 2 Z^T \vec{e} + 2 Z^T Z \Delta \vec{c}_1 \tag{3.65}$$

or

$$\Delta \vec{c}_1 = -(Z^T Z)^{-1} Z^T \vec{e} \tag{3.66}$$

assuming that $Z^T Z$ is nonsingular. The matrix

$$S = Z^T Z \tag{3.67}$$

is referred to as the regression matrix due to the similarity of this method to linear regression analysis. The normal equation for iteration, Eqn. (3.65), is linear in $\Delta \vec{c}$ only because function F was linearized and a quadratic criterion function was chosen.

Since

$$\vec{\nabla}\phi(\vec{c}_1) = \frac{\partial \phi}{\partial \vec{c}}\bigg|_{\vec{c} = \vec{c}_1} = \frac{\partial}{\partial \vec{c}} \vec{e}^T \vec{e}\bigg|_{\vec{c} = \vec{c}_1} = 2\left(\frac{\partial \vec{e}}{\partial \vec{c}}\right)^T_{\vec{c} = \vec{c}_1} \vec{e} = 2 \; Z^T \vec{e} \tag{3.68}$$

Eqn. (3.66) becomes

$$\Delta \vec{c}_1 = -\tfrac{1}{2} S^{-1} \; \vec{\nabla}\phi(\vec{c}_1) \stackrel{\Delta}{=} \vec{\beta}_1 \tag{3.69}$$

The new value for the parameter vector, $\bar{c}$, then is

$$\vec{c}_2 = \vec{c}_1 + \Delta \vec{c}_1 \tag{3.70}$$

upon which a new iteration may then be initiated. This procedure is referred to as the "Gauss-Newton" iteration method [17].

As mentioned previously, Eqn. (3.69) is based on the assumption that linearizing function F, Eqn. (3.59), is valid. Since, in fact, this may be completely invalid, it is quite possible that the sequence of parameter vector estimates, given by Eqn. (3.70), will not converge to $\vec{c}_0$. The necessary and sufficient conditions for the convergence of the Gauss-Newton procedure may be derived; however, the test is generally complicated enough that in practice one simply computes $\phi(\vec{c}_i + \vec{\beta}_i)$ at each step to see if an improvement results. It can be shown that the Gauss-Newton iteration always converges when binary scale factor adjustment is used [17]. When this technique is used, $\vec{c}_{i+1}$ is found from

$$\vec{c}_{i+1} = \vec{c}_i + 2^{-k} \; \vec{\beta}_i \tag{3.71}$$

where k is the first non-negative integer which reduces ϕ. However, experimental results show that the rate of convergence can be quite slow. For this reason, the "modified" Gauss-Newton procedure will not be used. The Gauss-Newton iteration enjoys its greatest success as a terminal iterative technique, where the current parameter vector is "close" to its minimizing value.

3.3.4 Newton-Raphson Iteration

When the Gauss-Newton iteration fails to give a reduced value for the criterion function, ϕ, then direct gradient techniques may be appropriate. This eliminates the necessity of inverting the matrix, S, and also makes it possible to handle parameter range constraints in a straightforward manner.

The gradient technique to be employed is the method of steepest descent, in which case the parameter change vector, Δc, is directly proportional to the negative gradient of the criterion function, ϕ.

$$\Delta \vec{c}_i = - k \ \vec{\nabla} \phi(\vec{c}_i) \tag{3.72}$$

where

$$k > 0 \tag{3.73}$$

Thus, $\Delta \vec{c}_i$ is in the direction of the greatest rate of decrease of ϕ. The next parameter vector estimate then becomes

$$\vec{c}_{i+1} = \vec{c}_i + \Delta \vec{c}_i \tag{3.74}$$

which may then be used to perform another iteration.

Before Eqn. (3.72) can be utilized, it is necessary to choose some value for the scale factor, k. The "Newton-Raphson" method may be used to obtain a value for k. Essentially it is based on taking the linear portion of the Taylor series expansion of the criterion function, ϕ, and extrapolating this to zero. More precisely, suppose that ϕ is a sufficiently smooth function of $\vec{c}$ such that it may be represented locally by the Taylor series

$$\phi(\vec{c} + \Delta \vec{c}) = \phi(\vec{c}) + \vec{\nabla} \phi^T \Delta \vec{c} + 0(\Delta \vec{c}^2) \tag{3.75}$$

where $0(\Delta \vec{c}^2)$ represents all the terms in the series which are quadratic or higher order in $\Delta \vec{c}$. Then for small $\Delta \vec{c}$, $0(\Delta \vec{c}^2)$ may be ignored, and

$$\phi(\vec{c} + \Delta \vec{c}) = \phi(\vec{c}) + \vec{\nabla} \phi^T \Delta \vec{c} \tag{3.76}$$

or

$$\phi(\vec{c} + \Delta \vec{c}) = \phi(\vec{c}) - k \ \left| \vec{\nabla} \phi \right|^2 \tag{3.77}$$

using Eqn. (3.72). Extrapolating ϕ to zero, for which $k = k^0$, gives

$$0 = \phi(\vec{c}_i) - k_i^0 \ \left| \vec{\nabla} \phi(\vec{c}_i) \right|^2 \tag{3.78}$$

or

$$k_i^0 = \frac{\phi(\vec{c}_i)}{\left| \vec{\nabla} \phi(\vec{c}_i) \right|^2} \tag{3.79}$$

The corresponding parameter change vector, $\Delta \vec{c}_i$, at each state of iteration is then, from Eqn. (3.72),

$$\Delta \vec{c}_i^{\,0} = - \frac{\phi(\vec{c}_i) \quad \vec{\nabla}\phi(\vec{c}_i)}{\left| \vec{\nabla}\phi(\vec{c}_i) \right|^2} \tag{3.80}$$

It may well be the case that iteration based on Eqn. (3.80) will give a value for $\phi(\vec{c} + \Delta \vec{c})$ which is larger than $\phi(\vec{c})$. This simply means that $\Delta \vec{c}$ is so large that linear extrapolation of ϕ to zero is, in fact, invalid. Eqn. (3.75) guarantees that for some $0 \leq k \leq k^0$ the criterion function, ϕ, will be reduced, however. Thus it is desired to find some $k = k^*$ such that

$$\begin{aligned} &\text{Min } \phi(\vec{c}_i - k_i \quad \vec{\nabla}\phi(\vec{c}_i)) = \phi(\vec{c}_i - k_i^* \quad \vec{\nabla}\phi(\vec{c}_i)) \\ &k > 0 \end{aligned} \tag{3.81}$$

The next parameter vector estimate is then

$$\vec{c}_{i+1} = \vec{c}_i - k_i^* \quad \vec{\nabla}\phi(\vec{c}_i) \tag{3.82}$$

This iteration scheme is called the "optimum gradient method".

It is, of course, not feasible to search over all values of k on a computer, but it is quite feasible to perform a binary search over the range

$$0 \leq k \leq \frac{\phi}{\left| \vec{\nabla}\phi \right|^2} = k^0 \tag{3.83}$$

which may be considered a "suboptimum gradient method". Assuming that ϕ is continuous and $\vec{\nabla}\phi \neq \vec{0}$, Eqn. (3.75) guarantees that there exists an n such that

$$\phi(\vec{c}_i - \frac{1}{2^n} k_i^0 \quad \vec{\nabla}\phi) < \phi(\vec{c}_i) \tag{3.84}$$

and therefore a binary search procedure always produces a convergent sequence of values for ϕ. A simple algorithm may be constructed to find the minimizing value for n as follows. First of all, compute $\Delta \vec{c}_i^{\,0}$ from Eqn. (3.80). Then, for $n = 0, 1, \cdots$, evaluate

$$\phi_i^n = \phi(\vec{c}_i + \frac{1}{2^n} \Delta \vec{c}_i^{\,0}) \tag{3.85}$$

Once a value for n is reached, say n = m, such that

$$\phi_i^{m-1} > \phi_i^m < \phi_i^{m+1} \tag{3.86}$$

and

$$\phi_i^m < \phi_i \tag{3.87}$$

then take for the new value of $\vec{c}$

$$\vec{c}_{i+1} = \vec{c}_i + \frac{1}{2^m} \Delta\vec{c}_i^0 \tag{3.88}$$

If Eqn. (3.87) is not satisfied, continue increasing n until Eqn. (3.86) is again satisfied and then check Eqn. (3.87) once again. Continue this until both Eqn. (3.86) and (3.87) are satisfied. Eqn. (3.88) is then the new value for $\vec{c}$.

A further refinement may be incorporated into this algorithm by fitting a quadratic function to the points ϕ_i^{m-1}, ϕ_i^m and ϕ_i^{m+1}. Letting $\phi_i^{m-1} = \phi_0$, $\phi_i^m = \phi_1$ and $\phi_i^{m+1} = \phi_2$, it is straightforward to show that the minimum of this fitted quadratic function occurs at

$$k_{quad} = \frac{3}{4} k_i^* \frac{\phi_2 - 5\phi_1 + 4\phi_0}{\phi_2 - 3\phi_1 + 2\phi_0} \tag{3.89}$$

where $$k_i^* = \frac{1}{2^m} k_i^0$$

If $$\phi(\vec{c}_i - k_{quad} \vec{\nabla}\phi) < \phi(\vec{c}_i - k_i^* \vec{\nabla}\phi) \tag{3.90}$$

then the parameter change vector is

$$\Delta\vec{c}_i = - k_{quad} \vec{\nabla}\phi(\vec{c}_i) \tag{3.91}$$

Otherwise,

$$\Delta\vec{c}_i = - k_i^* \vec{\nabla}\phi(\vec{c}_i) \tag{3.92}$$

3.3.5 Range Constraints

It is necessary to place constraints on the parameters since we assume the landmark or ellipse to be in the field of view. These are

range constraints, in which each component of the parameter vector is independently restricted to lie within some specified interval on the number scale. Thus, each component, c_i, must satisfy

$$a_i \leq c_i \leq b_i \qquad (3.93)$$

where a_i and b_i are the lower and upper limits of the allowed range, respectively. This then means that the parameter vector, $\vec{c}$, which minimizes the criterion function, ϕ, must be in a hypercube in parameter space.

Considering the problem at hand, one notes that in order for Eqn. (3.2) to represent an ellipse it is necessary that e_1 and e_2 (or c_1 and c_2, respectively) be positive. Likewise, the finite field of view of the optical equipment places constraints on e_1 and e_2 as well as the translations A and B (or c_3 and c_4, respectively). Since an ellipse is symmetric about its two axes, the rotation angle, $\theta = c_5$, may be constrained to lie in the first quadrant.

Since the gradient-descent method discussed earlier is valid only on the interior of the 5-dimensional contstraint region, R, it is necessary to use a different strategy when a constraint boundary is encountered during a gradient-descent. The method to be used is called the gradient-projection method. If a constraint boundary should be encountered, this method projects the gradient onto the constraint surface and then travels in the negative direction of the projected gradient until a minimum for ϕ is found. The actual minimum for ϕ may either be located on the interior of R or on a constraint boundary of R. In the latter case the projection of the gradient will have all of its components equal to zero at the final iteration.

The mechanization of the gradient-projection method may be performed in three steps.

1. Check each component of the current parameter vector estimate, $\vec{c}$, to see whether it is within the allowed range or if it lies at the lower or upper end of the range.

2. If any component lies on either extreme of its range, and if the negative of the corresponding gradient component points out of the constraint region, then set this component of the gradient equal to zero. Leave all other components of the gradient at their true value.

3. The resulting vector, $\vec{\nabla}\phi_p$, is the desired projected gradient.

In order to find the optimum step size, it is necessary to find the maximum scale factor which can be applied to $-\vec{\nabla}\phi_p$ without violating a range constraint. To this end, suppose that $\frac{\partial \phi_p}{\partial c_j}$ is positive. This means that c_j can be reduced in value without violating a range constraint. Let k_j^0 be the largest scale factor that can be applied to the negative j^{th} gradient component without violating the j^{th} range constraint. Then k_j^0 satisfies

$$c_j - k_j^0 \frac{\partial \phi_p}{\partial c_j} = a_j \tag{3.94}$$

which gives

$$k_j^0 = \frac{c_j - a_j}{\frac{\partial \phi_p}{\partial c_j}} \tag{3.95}$$

Likewise, for the negative components of $\vec{\nabla}\phi_p$ it follows that

$$k_j^0 = \frac{c_j - b_j}{\frac{\partial \phi_p}{\partial c_j}} \tag{3.96}$$

The maximum scale factor, k_0, is found from

$$k_0 = \underset{j}{\text{Min}} \{k_j^0\} \tag{3.97}$$

where the k_j^0 are defined only for the non-zero components of $\vec{\nabla}\phi_p$.

The maximum step size for the parameter change vector now becomes

$$\Delta\vec{c}_i = - k_0 \vec{\nabla} \phi_p \tag{3.98}$$

This value may then be used as the maximum step size for the binary search procedure discussed earlier.

3.3.6 Global Optima

Both the Gauss-Newton method and the Newton-Raphson method are suited for determining local minima since they make use only of local information. If more than one minimum is contained within the constraint region, R, then it is desirable to find the smallest of all of these minima. Such a minimum is called a global minimum. Of course, the only way to find the global minimum with certainty is to exhaustively search the entire constrained parameter space. Since this is not feasible or practical on a computer, one must choose some method whereby a given confidence level is attained that the minimizing parameter vector obtained is associated with a value of ϕ which is smaller than some specified per cent of the points in R. Such a method is that of uniform random searching in which parameter vectors are chosen at random (with a uniform distribution for each component) and their corresponding criterion functions are evaluated. The parameter vector, $\vec{c}$, being associated with the smallest value for ϕ is then used to initiate a local minimization [17].

CHAPTER IV

RECOGNITION OF ELLIPTICAL PLANAR PATTERNS

4.1 Introduction

This chapter is concerned with the recognition of elliptical planar patterns. By the term "recognition" it is meant that the two minimization techniques which are discussed in Chapter III are employed to estimate the five parameters associated with an ellipse having arbitrary size and shape, as well as arbitrary position (translation and rotation) in the planar field of view.

Section 4.2 discusses the statement of the problem and the general approach which is to be pursued, while section 4.3 discusses the various parameters which are associated with the implementation of the two minimization schemes.

The results which were obtained from the two minimization schemes are discussed in section 4.4, and a summary of the advantages and disadvantages of the two methods is contained in section 4.5.

4.2 Statement of Problem

The parameter estimation schemes were first applied to the recognition of elliptical patterns. Elliptical patterns were selected first because they are a somewhat complex pattern and yet their boundary may be represented analytically. Furthermore, most of the ground work for the estimation of the parameters of an ellipse has been laid in Chapter III.

As was discussed in Chapter III, an ellipse, located in a plane, may be fully characterized by five parameters. Two parameters, e_1 and e_2, are necessary to specify the size and shape of an ellipse, while three parameters are necessary to specify its position and orientation in the plane. Figure 7 shows a typical ellipse which has been translated and rotated with respect to the reference x, y-coordinate system.

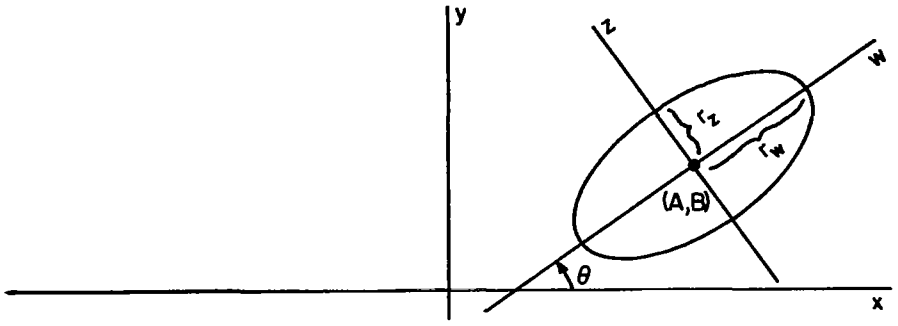


Fig. 7--Parameters of an ellipse in the x, y-reference frame.

With respect to the w, z-coordinate system, this ellipse may be expressed analytically by Eqn. (4.1)

$$e_1 w^2 + e_2 z^2 = 1 \tag{4.1}$$

where

$$e_1 = \frac{1}{r_w^2} \quad \text{and} \quad e_2 = \frac{1}{r_z^2} \tag{4.2}$$

The parameters r_w and r_z are respectively called the w-axis radius and the z-axis radius of the ellipse. The parameters which are actually estimated are e_1 and e_2, which are related to r_w and r_z by Eqn. (4.2). The x and y-translation parameters are denoted by A and B, respectively, and the rotation parameter is denoted by θ. These parameters are all shown in Figure 7.

The parameter vector, $\vec{c}$, which characterizes an ellipse is given by Eqn. (4.3).

$$\vec{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ A \\ B \\ \theta \end{bmatrix} \tag{4.3}$$

It is, then, the intent of this chapter to determine the feasibility of recognizing an elliptical planar pattern by estimating its associated parameter vector $\vec{c}$, and to determine whether the one step minimization method or the iterative minimization scheme does the better job of performing this parameter estimation task.

4.3 Implementation of the Parameter Estimation Schemes

To simulate an ellipse in the field of view, points which lie on the boundary of an ellipse are artificially generated by the subroutine denoted by DATA. The logic by which this subroutine selects the data points is discussed in Appendix I. After the subroutine DATA is provided with parameter vector $\vec{c}_o$, it generates data points which lie on the boundary of an ellipse which is characterized by $\vec{c}_o$. This parameter vector was arbitrarily chosen to be

$$\vec{c}_o = \begin{bmatrix} 1.00 \\ 0.25 \\ 1.00 \\ -2.00 \\ 0.50 \end{bmatrix} \qquad (4.4)$$

This corresponds to an ellipse which has a w-radius and a z-radius equal to 1.0 and 2.0, respectively. In addition, the ellipse has been translated one unit in the x-direction and two units in the negative y-direction, and rotated 0.5 radians.

Thus, in the absence of measurement noise, one would expect the estimate for $\vec{c}$ to be exactly $\vec{c}_o$.

While the parameter vector $\vec{c}_o$ was held fixed, two other parameters were varied to determine their effect on the accuracy of the parameter estimation schemes.

One of these variable parameters was the number of data points which were used to represent the boundary of the ellipse. Ten data points were chosen for a sparse distribution of points on the boundary, while one hundred data points were chosen for a dense distribution of points on the boundary. Intermediate values for the number of data points were chosen as 20 and 50.

The other variable parameter was the amount of noise which was added to the data points to simulate the effect of measurement noise or other errors. The noise samples, which are generated on the digital computer, have a gaussian distribution. The mean and standard deviation of these noise samples may be independently specified. In all cases the mean was chosen to be zero, while the standard deviation was either 0.0, 0.1, 0.2, 0.3, 0.4, or 0.5. The maximum value for the standard deviation of the noise sample, 0.5, was one-half of the z-radius of the noiseless ellipse. Noise samples having a standard deviation larger than 0.5 result in the data points having such a large scatter that they no longer even remotely resemble the boundary of an ellipse. In fact, physical systems which correspond to the higher values of standard deviation (0.3-0.5)

would have limited practical utility, but it is of interest to investigate the reliability of the parameter estimation schemes for high noise levels, and to develop bounds on the performance of such systems.

The pattern which is to be recognized is required to lie within some bounds since in a physical situation the optical system would have a finite field of view. The field of view was arbitrarily chosen to be a square measuring eight units on a side. The boundary of the ellipse which is characterized by the parameter vector $\vec{c}_o$ given by Eqn. (4.4) lies entirely within this field of view.

A remark should be made at this point. If the noise which is added to the data points has a large standard deviation, it is possible that some of the resulting noisy data points will fall outside of the field of view. When this situation arises, those noisy data points which fall outside of the field of view are still regarded as valid data points in the simulation. Physically, in an actual landmark tracking or automatic docking situation, noise may be classified into two general categories. The first category consists of noise associated with measurement errors. These include grid quantization errors, detector or sensor errors, and transmission errors. In any of these cases the coordinates of a data point (which is in the optical field of view) will be in error, and if the true data point is near the boundary of the field of view then it is possible that the noisy, or measured, data point will have coordinates which lie outside of the field of view. This situation is contrasted to the second-category into which noise may be classified, which may be termed "masking" noise for lack of a better name. This kind of noise corresponds to a case in which the field of view is partially covered with clouds or to a case in which the optical system is very badly out of focus. The sensor will be unable to detect the data points which are masked, or obscured, due to either of these situations, and therefore these data points are in essence, discarded. This masking noise has the effect, therefore, of shrinking the field of view.

Thus, it can be seen that the noise which is being simulated corresponds to measurement noise rather than "masking" noise.

The numerical values utilized in simulation experiments for the range constraints for the five ellipse parameters are as follows:

$$\begin{aligned} 0.0625 \leq\ & e_1 \leq 16.0 \\ 0.0625 \leq\ & e_2 \leq 16.0 \\ -4.0 \leq\ & A \leq 4.0 \\ -4.0 \leq\ & B \leq 4.0 \\ 0 \leq\ & \theta \leq 1.57 \end{aligned} \qquad (4.5)$$

These range constraints permit the fitted ellipse to have either of its radii range in size from 1/4 unit to 4 units (i.e., the maximum diameter is constrained to be no larger than the dimensions of the field of view). In addition, the center of the fitted ellipse is permitted to lie anywhere within the field of view, while the rotation angle is constrained to lie in the first quadrant due to the symmetry of an ellipse.

The estimation process has to be started with an arbitrary initial parameter vector, $\vec{c}_e$. The initial guess for the parameter vector was

$$\vec{c}_e = \begin{bmatrix} 0.5 \\ 1.3 \\ 0.7 \\ -2.5 \\ 0.8 \end{bmatrix} \qquad (4.6)$$

which corresponds to an ellipse having a w-axis radius and a z-axis radius equal to 1.414 and 0.877, respectively. The parameter vector $\vec{c}_e$ was chosen such that its components were "close" in value to the components of $\vec{c}_o$ and yet not so close as to make the estimation problem trivial.

After performing a local minimization using $\vec{c}_e$ as the initial estimate for $\vec{c}_o$, four more local minimizations are executed with the initial estimate in each case being found by the RANSER (random search) subroutine. [17] Thus, a total of five trial local minimizations are carried out. The number of random searches for each trial local minimization was set equal to 100.

4.4 Results

Both the one step minimization method and the iterative minimization scheme which are outlined in Chapter III were employed to estimate the parameters of the given ellipse. The results which were obtained by using these two schemes are shown in Tables 1 and 2, respectively. It should be pointed out that the data points are exactly the same for both minimization schemes, permitting a meaningful comparison to be made. The results which are tabulated in Table 2 are also shown pictorially in Figures 8, 9, 10, 11, 12, and 13. In these figures the ellipse having a solid line boundary corresponds to the parameter vector $\vec{c}_o$. The "+" symbols correspond to the noisy data points arising from the solid line boundary. The ellipse which is fitted to these noisy data points, and characterized by $\vec{c}_e$, is represented by the dashed line boundary. The x and y-radii correspond to the w-axis and z-axis radii, respectively.

A comparison of Tables 1 and 2 shows that for noise levels below $\sigma = 0.4$ the two minimization schemes produced results which were quite similar. Referring to these tables or to Figure 8 one notes that for noiseless data points the parameter vector is estimated precisely, that is, $\vec{c}_e = \vec{c}_o$. This is a criterion which any good recognition scheme should fulfill, of course.

Figure 9 shows the results which were obtained for $\sigma = 0.1$. Special note should be made concerning the accuracy with which the rotation component of the parameter vector was estimated. It is seen that the largest error is less than 3 degrees, while for three of the four cases this error is considerably less than one degree.

A brief comment concerning the expression for the error should be made at this point. In most instances it is more convenient and meaningful to express an error in percentage rather than absolute terms. Such is the case here. Since an ellipse is symmetric about both its vertical and horizontal axes, its angular position is unique only in the first quadrant, i.e., 0 to 90 degrees. The percentage error may then be defined as the ratio of the absolute error to 90 degrees. With the percentage error so defined, one can see that for $\sigma = 0.1$ the maximum error is approximately three percent for the rotation parameter, which is quite good considering the fact that the reference rotation angle is not constrained to be small.

The results for $\sigma = 0.2$ are shown in Figure 10. Here again one notes that the error in estimating the rotation parameter is quite good. In fact, ignoring the 10 data point case, the maximum error is still less than three percent. In the 10 data point case the error is approximately seven percent, which is still reasonable considering the scarcity of data points and the noise level. Another observation which can be made from both Figures 9 and 10 is that the estimates for the parameter vector become better as more data points are used, a situation which intuitively seems reasonable.

When the noise level reaches $\sigma = 0.3$, as shown in Figure 11, the fitted ellipses begin to differ from the reference ellipses to a larger, and perhaps unacceptable, extent. The scatter of the data points is such that an accurate fit cannot be realized by either of the parameter estimation schemes. However, it should be pointed out that the estimate for the rotation parameter is still respectable, except for the 10 data point case. In the other cases the maximum error in the rotation parameter estimate is less than nine percent, and for the 100 data point case this error is approximately three percent (for the iterative minimization scheme).

For higher noise levels (σ = 0.4 and 0.5) the one step minimization method runs into serious difficulties. Table 1 shows two instances in which the one step minimization method was unable to fit an ellipse to the data points. In both of these instances the estimate $\hat{c}_e$ was found to have one of its first two components a negative number. This means that the one step minimization method actually fit a hyperbola to the given data points rather than an ellipse.

For relatively high levels of noise (σ = 0.4 and 0.5) Table 2 shows that the iterative minimization scheme also exhibits an undesirable characteristic, that is, it has a tendency to select values for the first two components of $\hat{c}_e$ which are at the boundary of their respective range constraints. When this is the case the resulting fitted ellipse is actually a circle having a radius equal to four units, as shown in Figures 12 and 13. These figures indicate that the iterative minimization scheme has attempted to cluster all of the data points along a small portion of the boundary of the fitted ellipse (or circle), with approximately one half of the data points on either side of the fitted ellipse's boundary.

4.5 Summary

This chapter has investigated the merits of "recognizing" a planar elliptical pattern, whose boundary points are given, by estimating the values for the five parameters which characterize an ellipse. The parameter estimation schemes which were employed are the two which were described in Chapter III, namely, the one step minimization method and the iterative minimization scheme. The ellipse which was to be recognized was permitted to have arbitrary size and shape, as well as arbitrary position and orientation so long as it was located within the specified field of view.

As was pointed out in Section 4.4, the two minimization schemes provided essentially the same results for the estimate of the parameter vector associated with the reference ellipse when the noise level was below σ = 0.4. For the lower noise levels (σ = 0.0, 0.1, and 0.2) these estimates were quite good, and special note was made concerning the accuracy with which the rotation parameter was estimated. Excluding the 10 data point case for σ = 0.2, the rotation parameter was never more than three percent in error, which is a remarkable result. Unfortunately, no other schemes exist presently with which these results can be compared.

For σ = 0.3 the estimate for the rotation parameter was still respectable, but the other parameters were not estimated accurately enough to yield a fitted ellipse which approximated the reference ellipse to an acceptable degree.

Both minimization schemes displayed undesirable characteristics for very high noise levels (σ = 0.4 and 0.5). The one step minimization method had a tendency to fit a hyperbola to the data points rather than an ellipse (characterized by a negative value for one of the first two components of the parameter vector) while the iterative minimization scheme had a tendency to fit a constrained ellipse to the data points ($r_w = r_z = 4.0$).

The fact that both minimization schemes failed to accurately estimate the parameters associated with the reference ellipse for high noise levels does not distract from their usefulness. In practice one would regard a system corresponding to σ = 0.3, 0.4 and 0.5 as having an unacceptable level of noise and hence would demand a better design for the system. Upon viewing Figures 5, 6, and 7 one sees that it would be very difficult, if not impossible, to develop a recognition scheme that could accurately recognize an ellipse from the given scatter of data points (this includes a human being as a "pattern recognizer").

As a minor point, it should be mentioned that the undesirable characteristics of the two minimization schemes (for high noise) which were previously mentioned can be corrected to some extent. The iterative minimization scheme can be improved if the range constraints on the first two components of $\vec{c}_e$ are further restricted after the data points become known. One simple procedure is to construct a rectangle, having sides parallel to the x, y-axes, that encloses all the data points and that has at least one data point lying on each of its sides. One would expect the fitted ellipse to have neither of its diameters larger than the diagonal of this "bounding" rectangle. Thus, the two radii are constrained to be no larger than one half of this diagonal, and so the lower bounds on the range constraints for the first two components of $\vec{c}_e$ are modified accordingly.

For the one step minimization method, constraints could be specified so that the fitted pattern is forced to be an ellipse. However, the simplicity of the one step minimization method involved the fact that it was an unconstrained minimization scheme. Since the unconstrained one step minimization method was unable to always fit an ellipse to the data points (and for reasons given in Chapter V) the iterative minimization scheme was considered the better scheme and was used for the recognition of elliptical patterns as well as for patterns that are not ellipses.

Another point which might be noted concerning the one step minimization method is that this scheme tends to estimate the larger

radius, r_x, much less accurately than the smaller radius, r_y. This can be seen in Table 1. This is not a property of the iterative minimization scheme, however, giving more support for its use.

TABLE 1: ELLIPSE PARAMETER ESTIMATES OBTAINED BY ONE STEP MINIMIZATION METHOD

$r_x = 2.0$ $r_y = 1.0$

σ	N	e_1	e_2	A	B	θ	r_x	r_y
0.0	10	0.250035	1.000063	0.999918	-1.999968	0.499943	2.000	1.000
0.0	20	0.249987	0.999831	0.999987	-2.000008	0.499964	2.000	1.000
0.0	50	0.249962	1.000428	1.000113	-2.000046	0.500288	2.000	1.000
0.0	100	0.250081	1.000146	0.999886	-1.999961	0.499766	2.000	1.000
0.1	10	0.226588	0.985690	1.133521	-2.017937	0.505674	2.101	1.007
0.1	20	0.247580	1.019186	0.986980	-2.036948	0.500796	2.010	0.991
0.1	50	0.230059	0.966594	1.061745	-2.000900	0.545271	2.085	1.017
0.1	100	0.240472	1.021042	0.960186	-1.995705	0.504009	2.039	0.990
0.2	10	0.250661	1.091854	1.046996	-1.926707	0.610062	1.997	0.957
0.2	20	0.205477	0.870532	1.019036	-2.167258	0.457618	2.206	1.072
0.2	50	0.245096	1.081746	1.029216	-2.026368	0.482127	2.020	0.961
0.2	100	0.222750	1.104007	0.993413	-2.054868	0.527154	2.119	0.952
0.3	10	0.122579	1.141891	0.510788	-2.504557	0.728425	2.856	0.936
0.3	20	0.241930	1.038012	1.264826	-1.812599	0.547318	2.033	0.982
0.3	50	0.187579	0.965065	1.005195	-2.067051	0.611433	2.309	1.018
0.3	100	0.141655	0.976113	0.911619	-2.123045	0.518779	2.657	1.012
0.4	10	0.156311	0.662240	0.965569	-1.958783	0.386650	2.529	1.229
0.4	20	0.037682	1.751601	0.351396	-2.735868	0.661454	5.151	0.756
0.4	50	0.029652	0.997244	3.404136	-1.132561	0.461635	5.807	1.001
0.4	100	*						
0.5	10	0.081115	0.644221	2.139103	-1.733687	0.542781	3.511	1.246
0.5	20	0.137709	0.967893	0.618680	-2.566473	0.499484	2.695	1.016
0.5	50	*						
0.5	100	0.075127	1.015188	1.068787	-2.383365	0.507140	3.648	0.992

*Unable to fit ellipse.

Note: Reference ellipse parameters are: $e_1 = 0.25$, $e_2 = 1.00$, $A = 1.00$, $B = -2.00$, $\theta = 0.50$

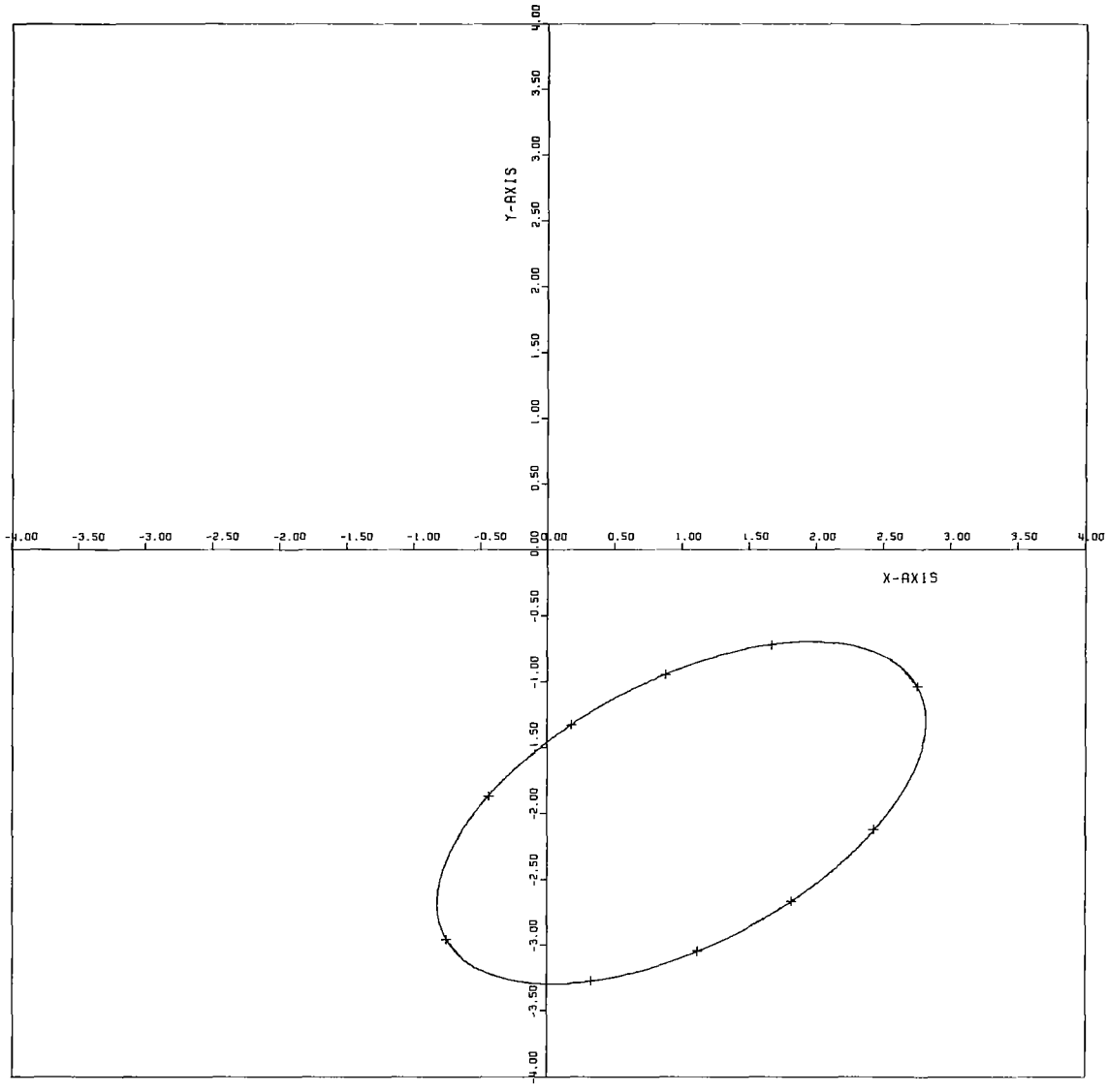
TABLE 2: ELLIPSE PARAMETER ESTIMATES OBTAINED BY ITERATIVE MINIMIZATION SCHEME

$r_x = 2.0$ $r_y = 1.0$

σ	N	e_1	e_2	A	B	θ	r_x	r_y
0.0	10	.25000	1.00000	1.00000	-2.00000	0.50000	2.000	1.000
0.0	20	.25000	1.00000	1.00000	-2.00000	0.50000	2.000	1.000
0.0	50	.25000	1.00000	1.00000	-2.00000	0.50000	2.000	1.000
0.0	100	.25000	1.00000	1.00000	-2.00000	0.50000	2.000	1.000
0.1	10	.23309	0.91828	1.12770	-2.00796	0.50372	2.071	1.044
0.1	20	.25481	0.97564	0.98513	-2.03016	0.50099	1.981	1.012
0.1	50	.24045	0.90386	1.05204	-1.99382	0.54868	2.039	1.052
0.1	100	.25103	0.95177	0.95558	-1.98589	0.50581	1.996	1.025
0.2	10	.26219	0.98635	1.04461	-1.91310	0.60783	1.953	1.007
0.2	20	.23523	0.65429	0.99513	-2.13235	0.46806	2.062	1.236
0.2	50	.28036	0.86192	1.02352	-1.98737	0.49220	1.889	1.077
0.2	100	.26151	0.86500	0.97935	-2.01866	0.54554	1.955	1.075
0.3	10	.17273	0.78372	0.56804	-2.39295	0.80087	2.406	1.130
0.3	20	.32107	0.62374	1.13235	-1.81673	0.61739	1.765	1.266
0.3	50	.23160	0.65681	0.99362	-1.99064	0.63811	2.078	1.234
0.3	100	.22484	0.60177	0.88051	-2.01215	0.54895	2.109	1.289
0.4	10	.16435	0.53921	0.80733	-1.93123	0.30713	2.467	1.362
0.4	20	.20117	0.74273	0.79921	-2.26838	0.70981	2.230	1.160
0.4	50	.06250	0.06250	-0.55830	1.00064	0.11353	4.000	4.000
0.4	100	.06250	0.06250	-1.12144	1.08820	0.12380	4.000	4.000
0.5	10	.10609	0.55384	1.85071	-1.87407	0.49956	3.070	1.344
0.5	20	.26380	0.44969	0.74672	-2.38593	0.64307	1.947	1.491
0.5	50	.06250	0.06250	-0.04203	1.33321	0.13402	4.000	4.000
0.5	100	.06250	0.06250	-1.35350	0.90174	0.14893	4.000	4.000

*Note: Reference ellipse parameters are: $e_1 = 0.25$, $e_2 = 1.00$, $A = 1.00$, $B = 2.00$, $\theta = 0.50$

FIELD OF VIEW



10 DATA POINTS
STANDARD DEVIATION = 0.0

REFERENCE ELLIPSE

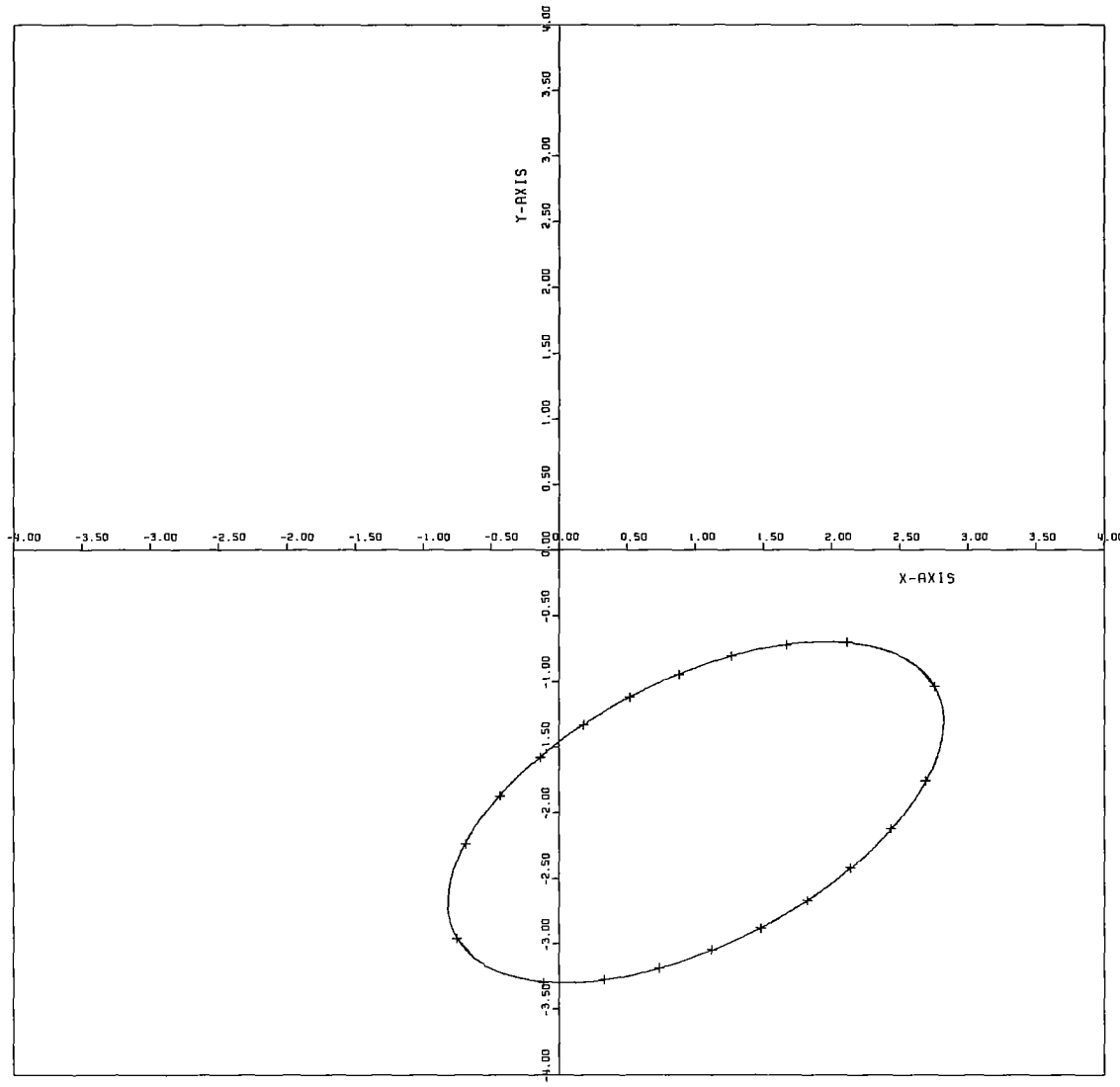
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

Fig. 8--Ellipses fitted to data points, $\sigma = 0.0$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.0

REFERENCE ELLIPSE

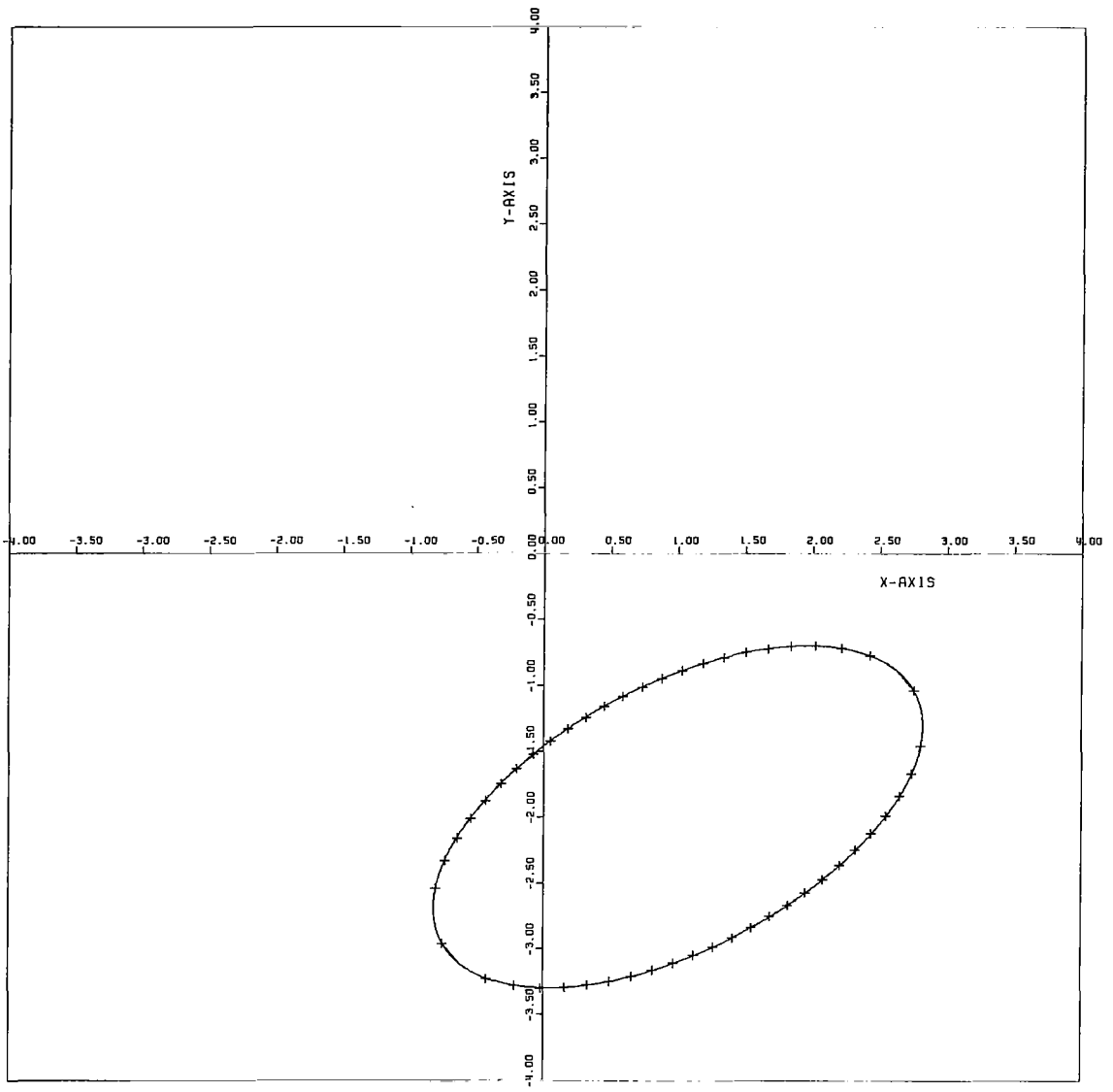
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

Fig. 8--Continued .

FIELD OF VIEW



50 DATA POINTS
STANDARD DEVIATION = 0.0

REFERENCE ELLIPSE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

Fig. 8--Continued .

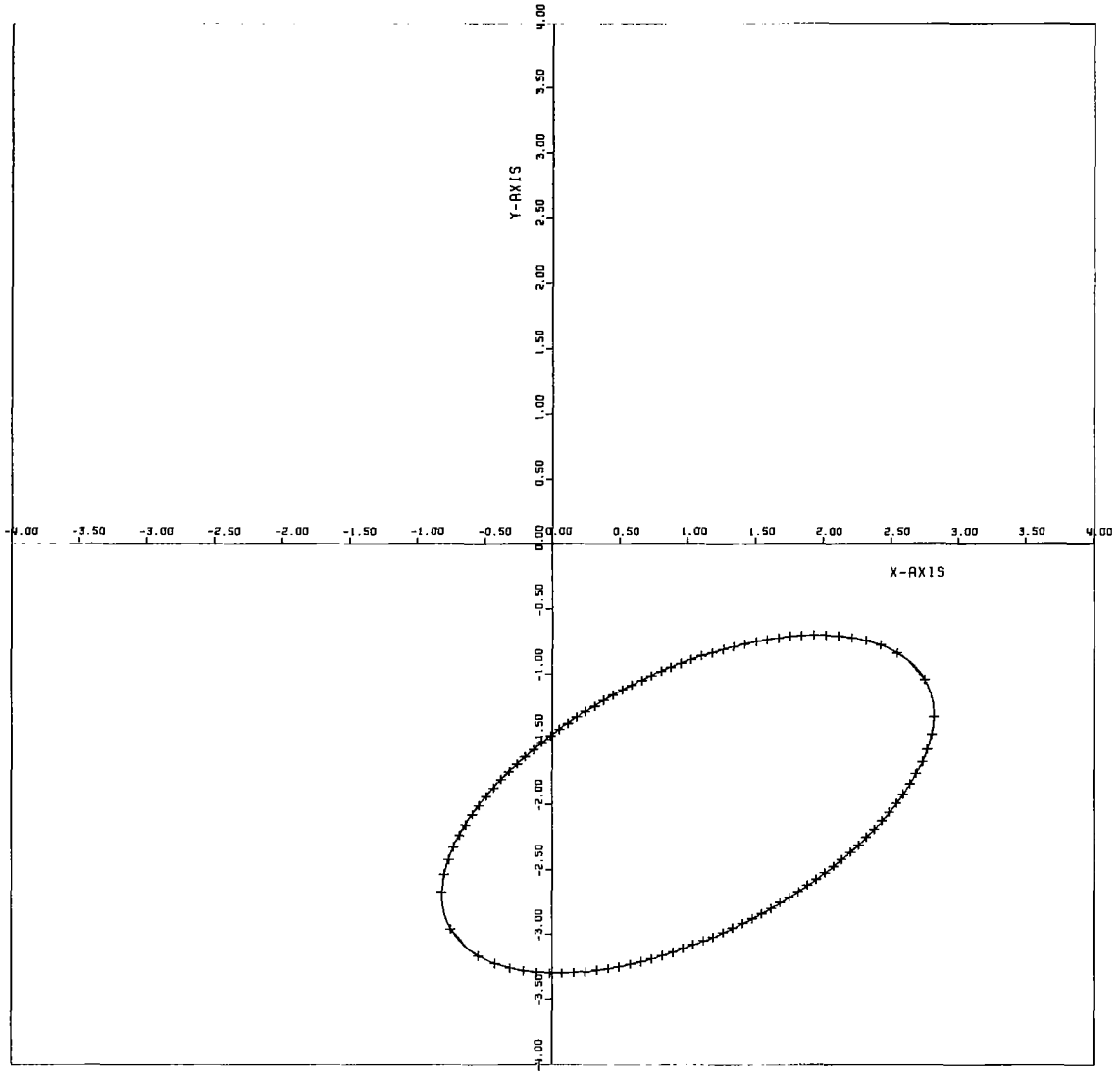
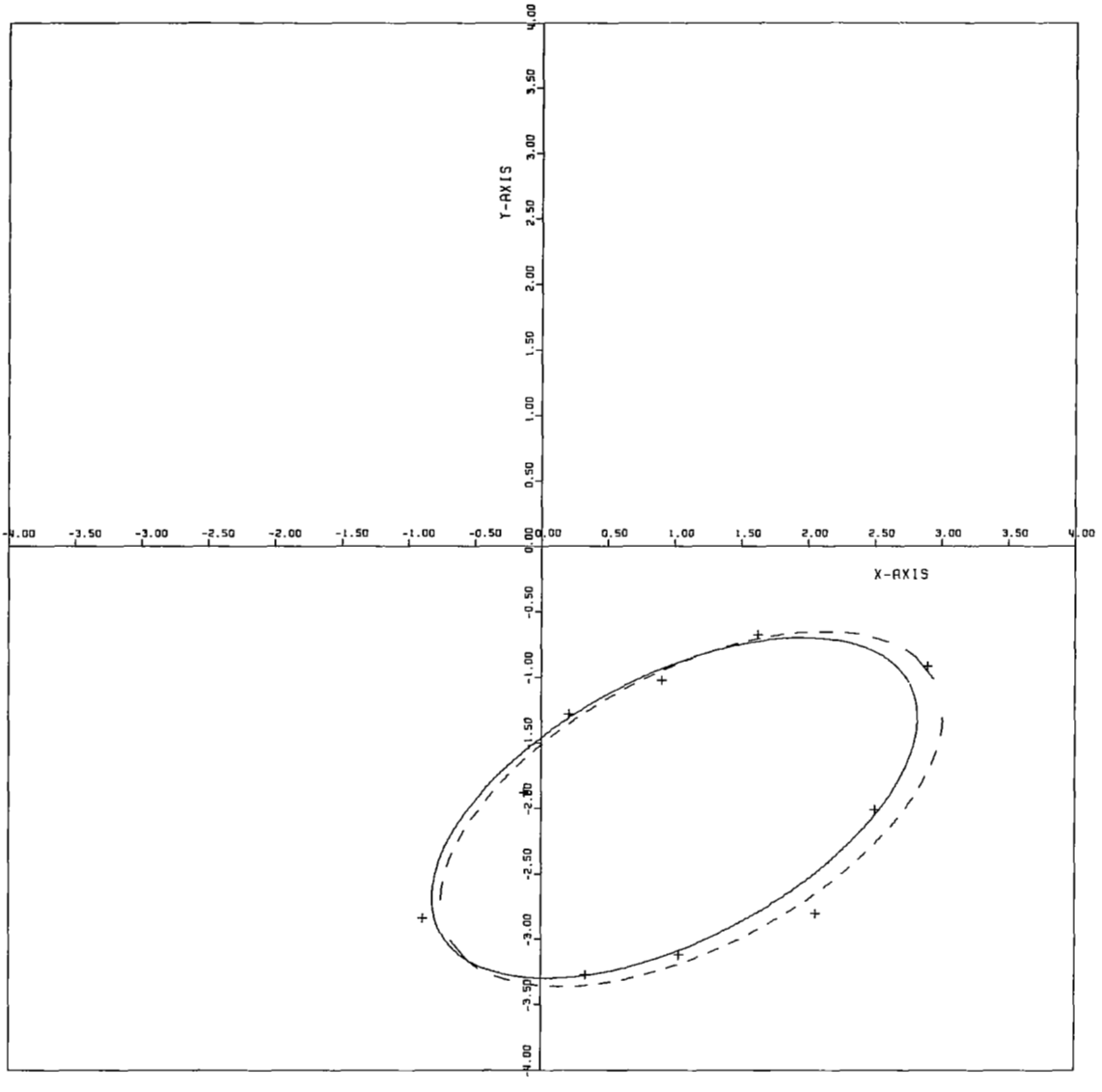


Fig. 8--Concluded .



10 DATA POINTS
STANDARD DEVIATION = 0.1

REFERENCE ELLIPSE

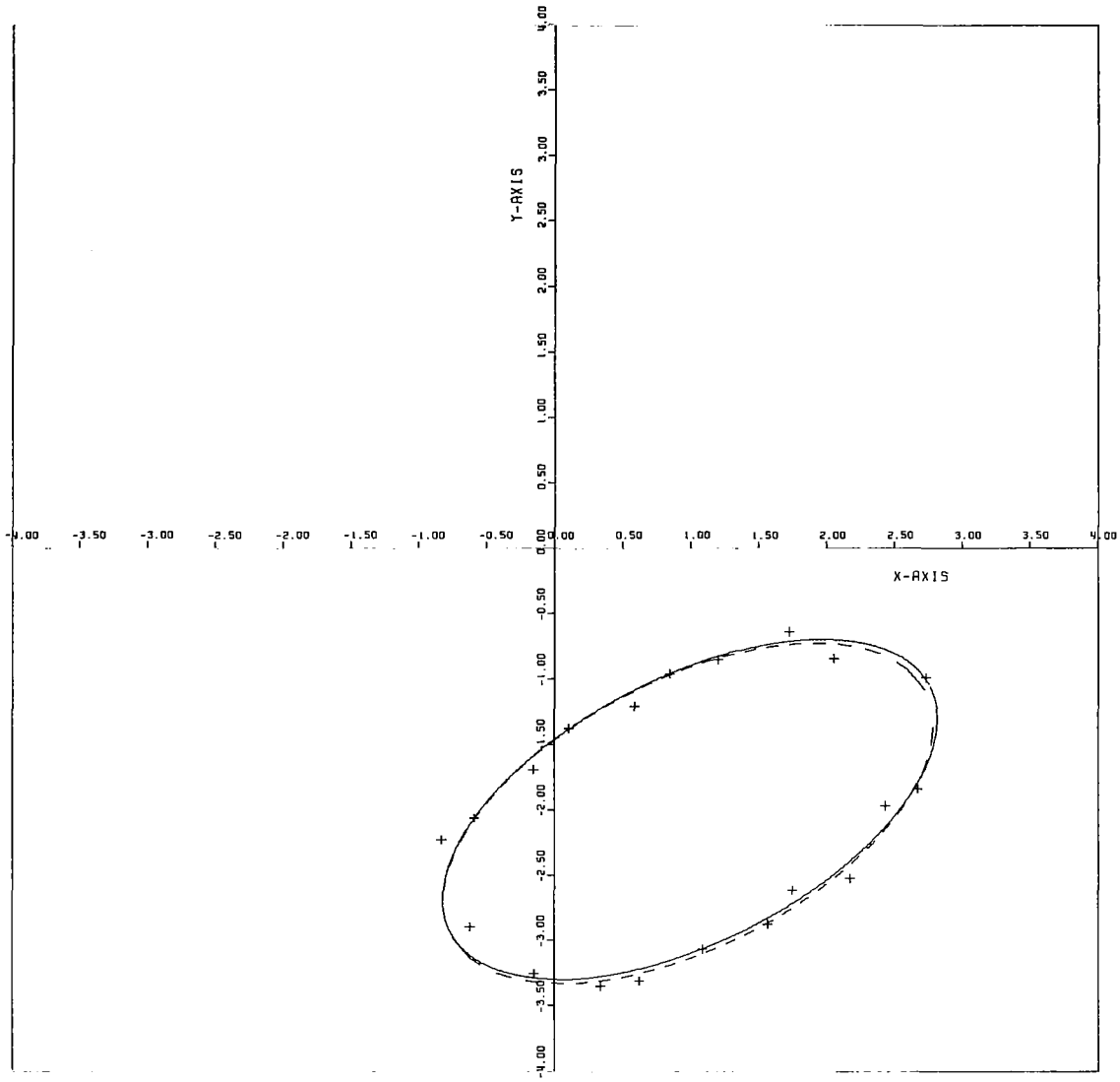
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.071
Y-RADIUS = 1.044
X-TRANSLATION = 1.128
Y-TRANSLATION = -2.008
ROTATION IN DEGREES = 28.861

Fig. 9--Ellipses fitted to data points, $\sigma = 0.1$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.1

REFERENCE ELLIPSE

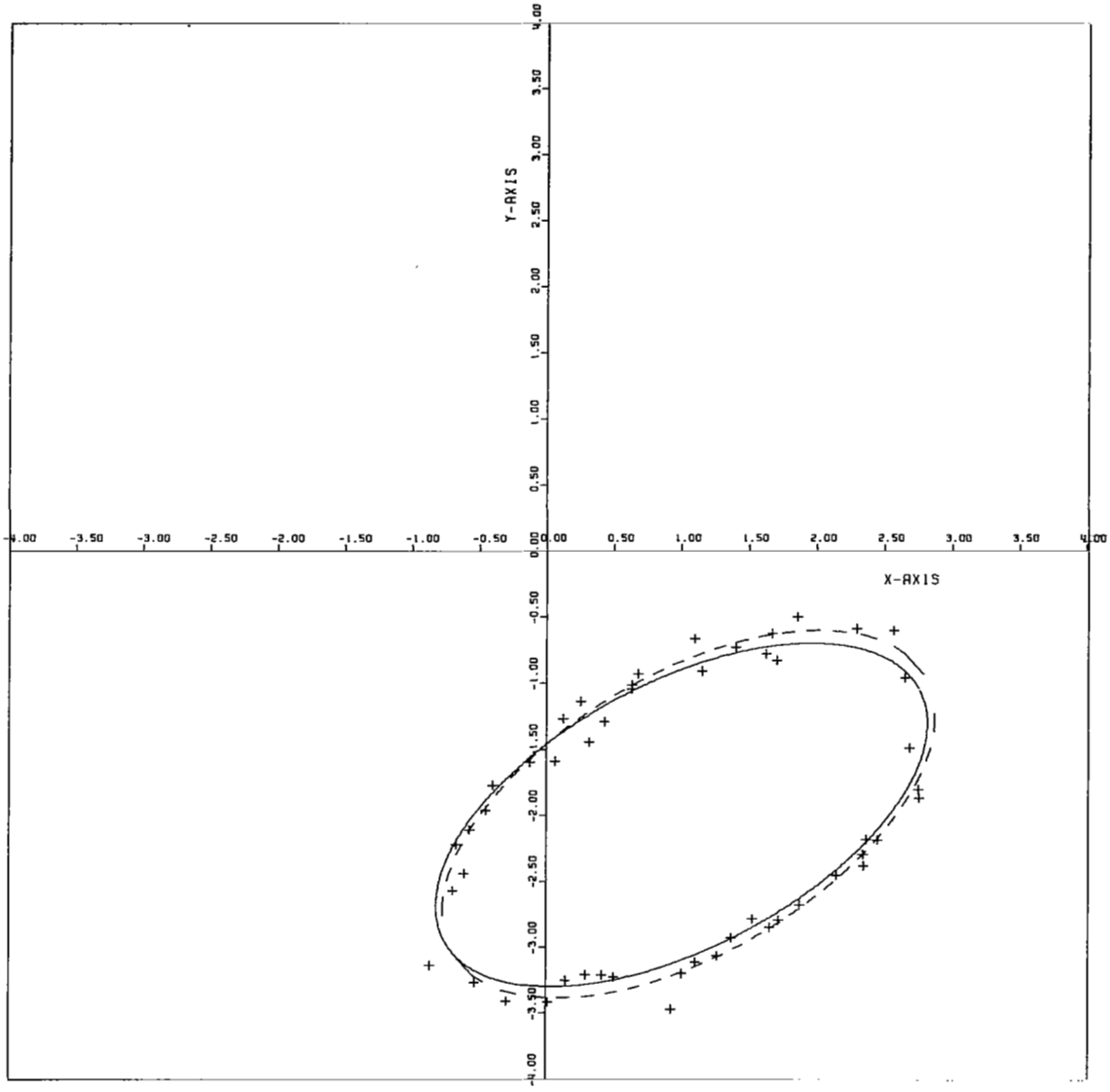
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 1.981
Y-RADIUS = 1.012
X-TRANSLATION = 0.985
Y-TRANSLATION = -2.030
ROTATION IN DEGREES = 28.705

Fig. 9--Continued .

FIELD OF VIEW



50 DATA POINTS
STANDARD DEVIATION = 0.1

REFERENCE ELLIPSE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.039
Y-RADIUS = 1.052
X-TRANSLATION = 1.052
Y-TRANSLATION = -1.994
ROTATION IN DEGREES = 31.437

Fig. 9--Continued .

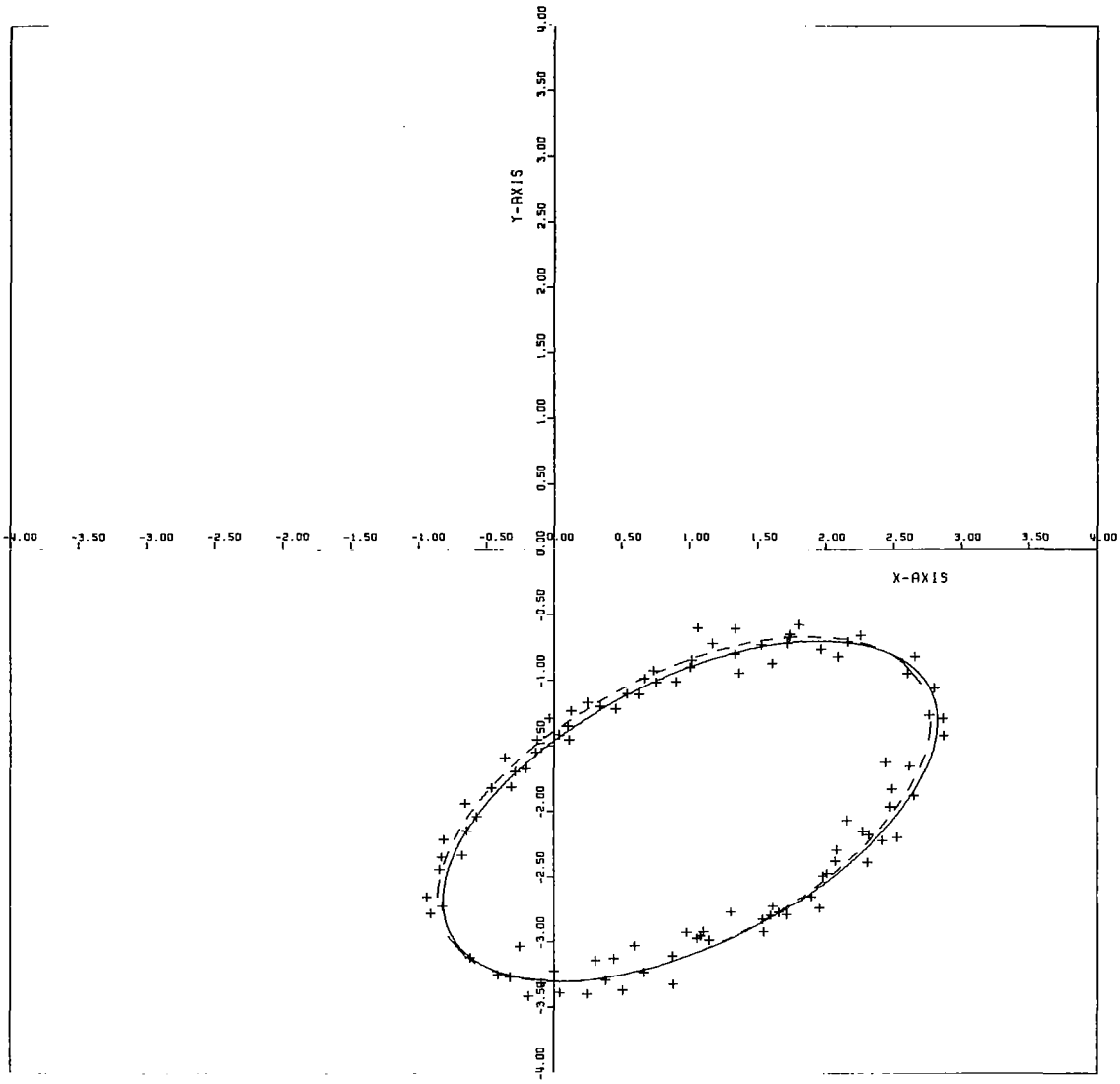
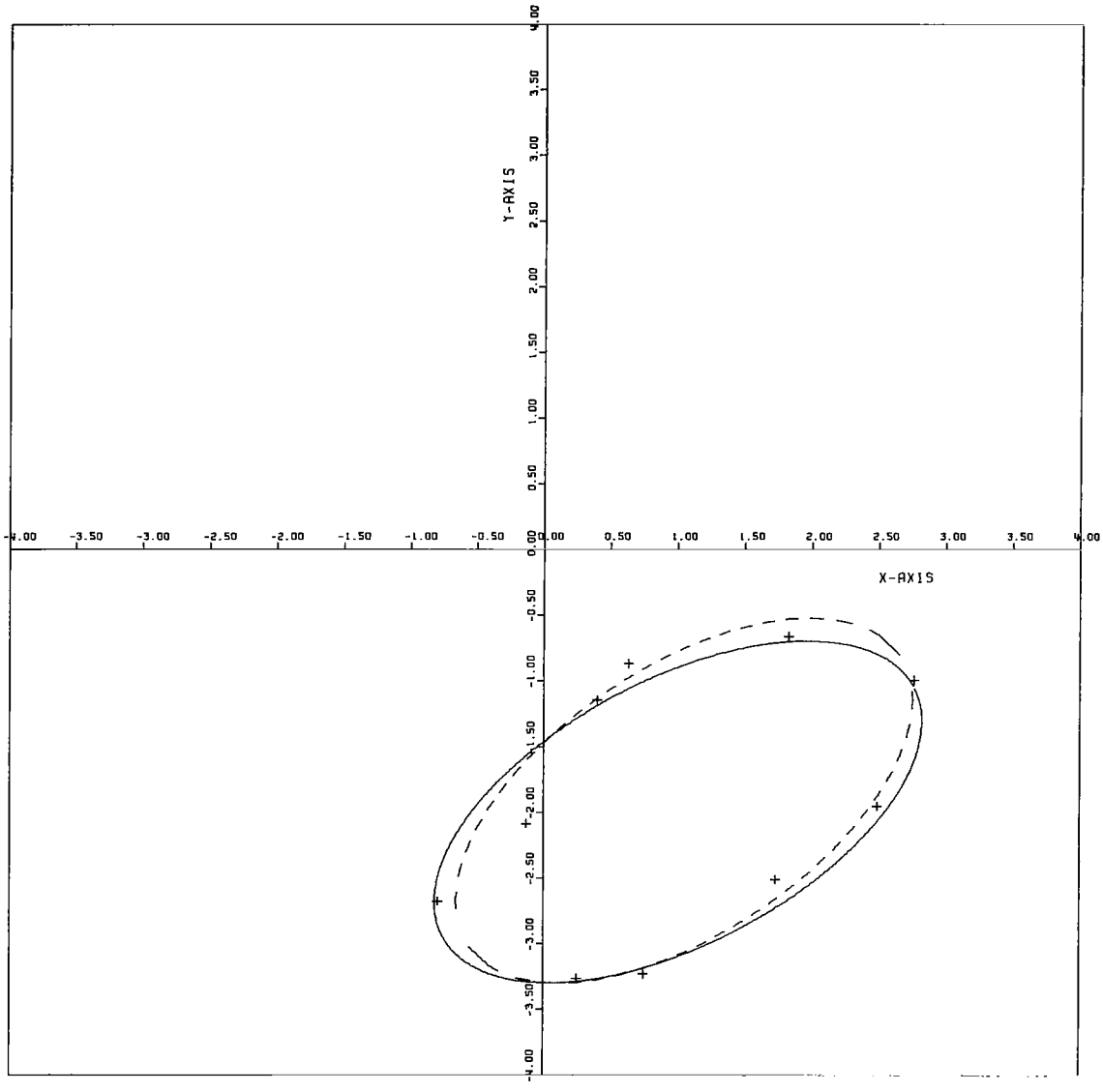


Fig. 9--Concluded .



10 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE ELLIPSE

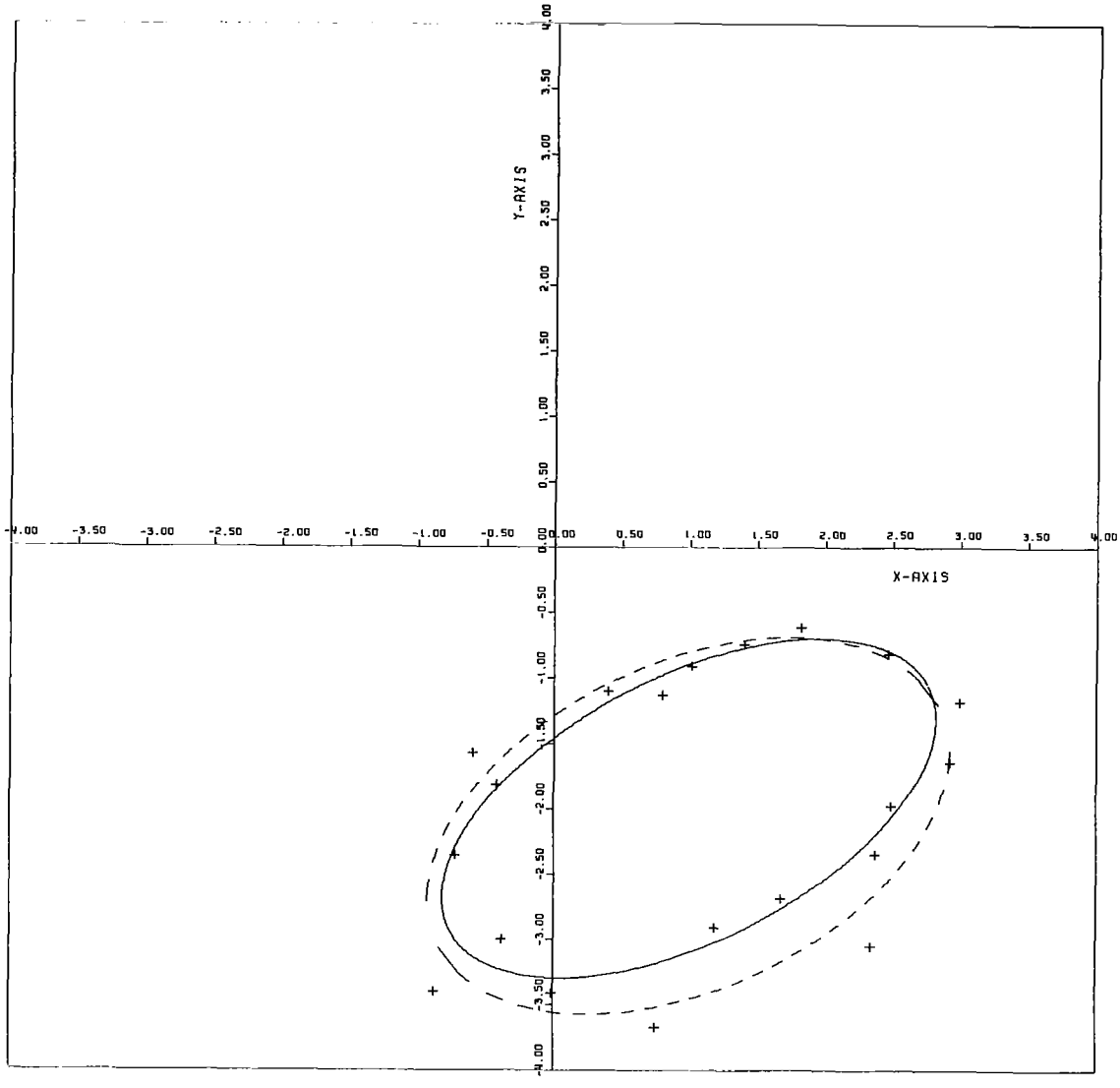
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 1.953
Y-RADIUS = 1.007
X-TRANSLATION = 1.045
Y-TRANSLATION = -1.913
ROTATION IN DEGREES = 34.826

Fig. 10--Ellipses fitted to data points, $\sigma = 0.2$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE ELLIPSE

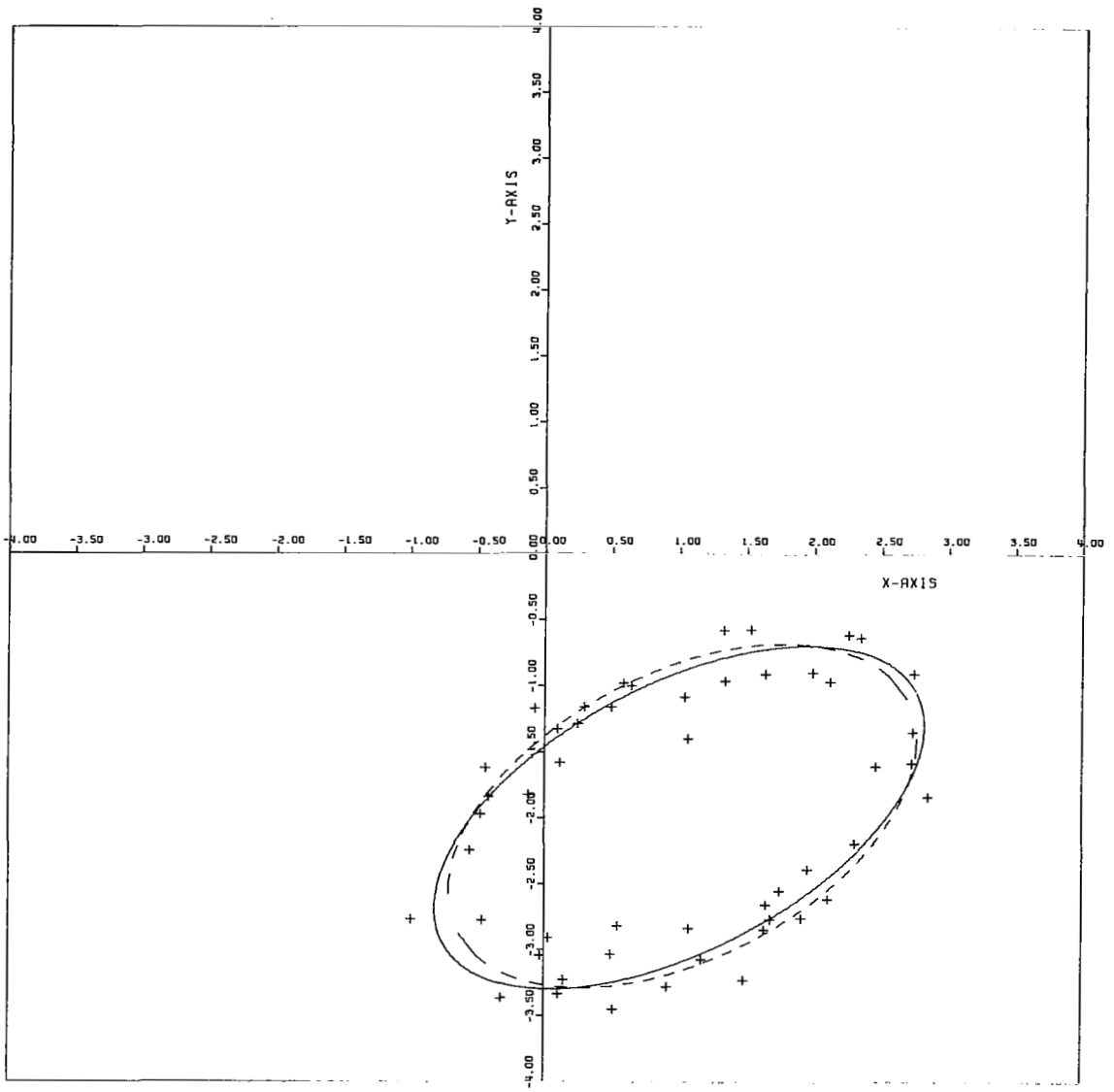
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.062
Y-RADIUS = 1.236
X-TRANSLATION = 0.995
Y-TRANSLATION = -2.132
ROTATION IN DEGREES = 26.818

Fig. 10--Continued .

FIELD OF VIEW



50 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE ELLIPSE

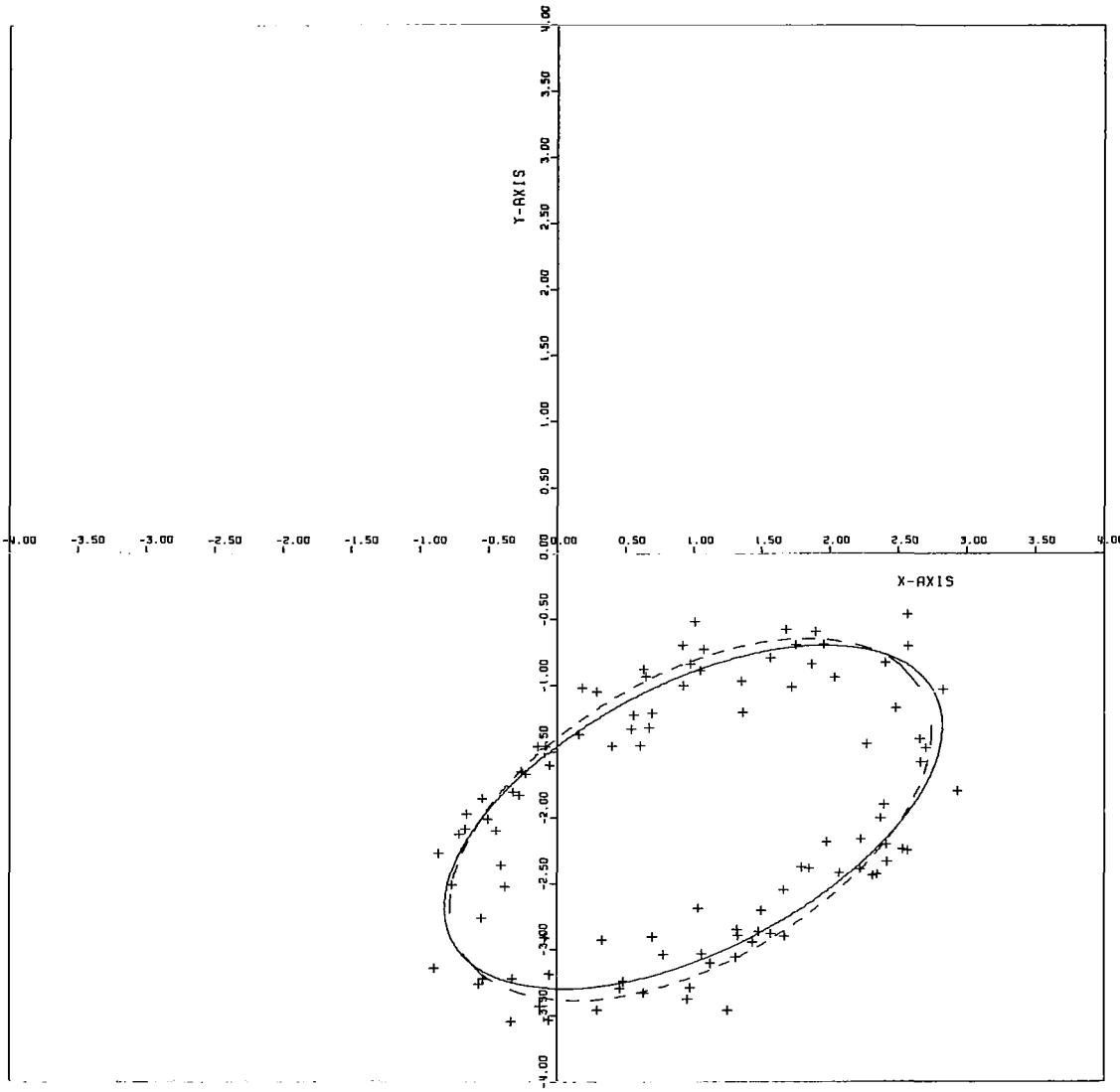
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 1.889
Y-RADIUS = 1.077
X-TRANSLATION = 1.024
Y-TRANSLATION = -1.987
ROTATION IN DEGREES = 28.201

Fig. 10--Continued .

FIELD OF VIEW



100 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE ELLIPSE

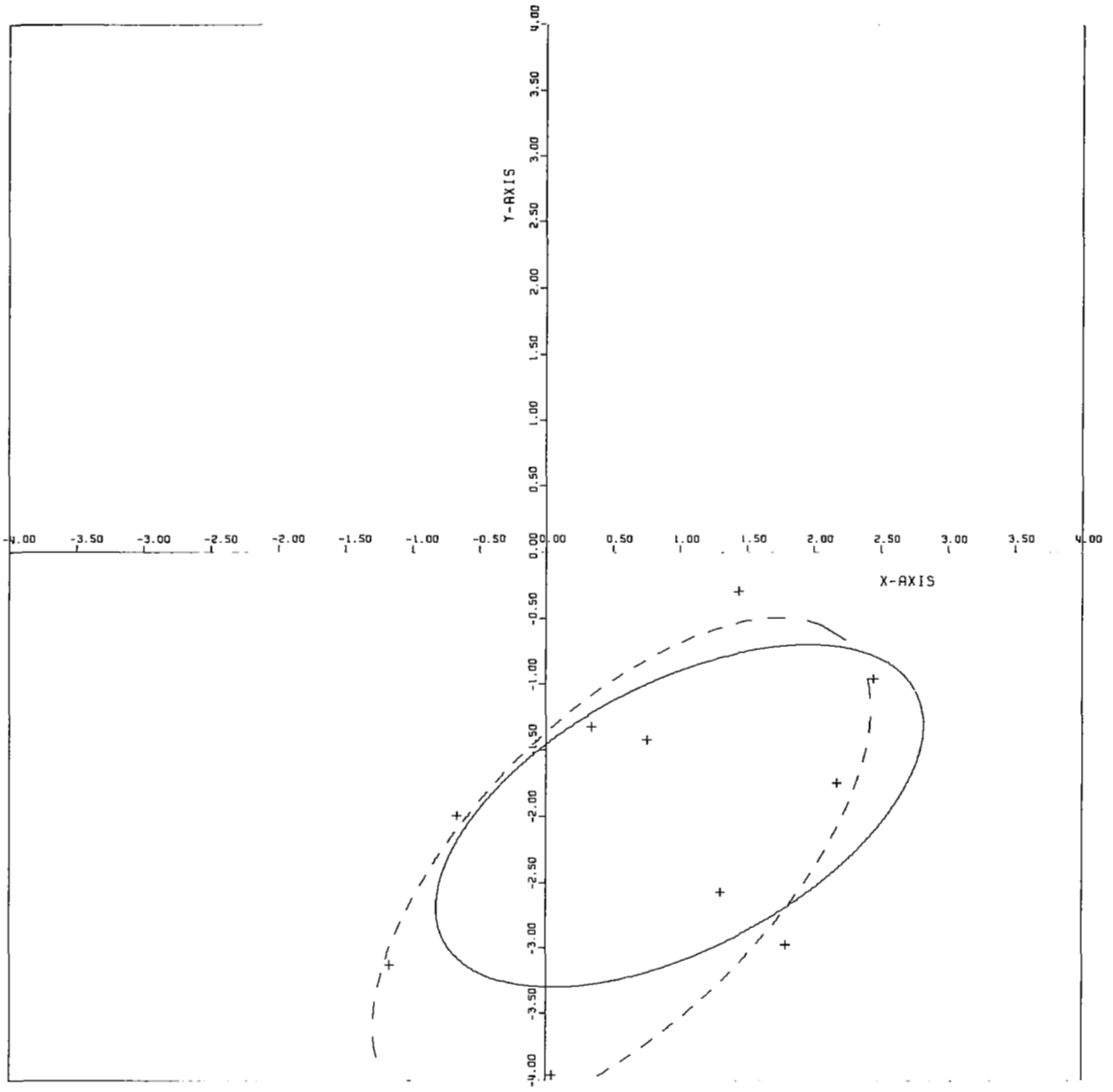
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 1.955
Y-RADIUS = 1.075
X-TRANSLATION = 0.979
Y-TRANSLATION = -2.019
ROTATION IN DEGREES = 31.257

Fig. 10--Concluded .

FIELD OF VIEW



10 DATA POINTS
STANDARD DEVIATION = 0.3

REFERENCE ELLIPSE

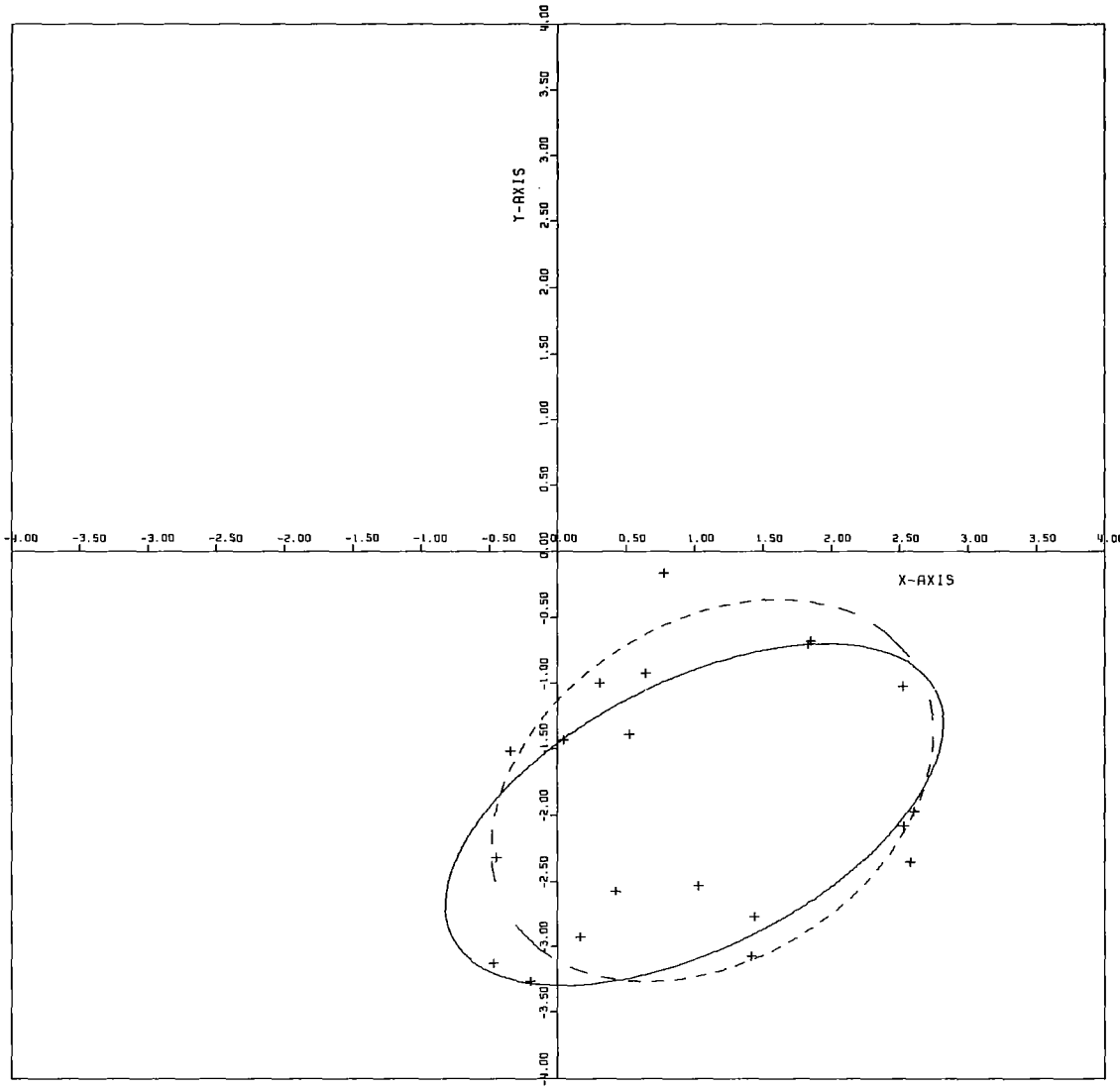
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.406
Y-RADIUS = 1.130
X-TRANSLATION = 0.568
Y-TRANSLATION = -2.393
ROTATION IN DEGREES = 45.887

Fig. 11--Ellipses fitted to data points, $\sigma = 0.3$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.3

REFERENCE ELLIPSE

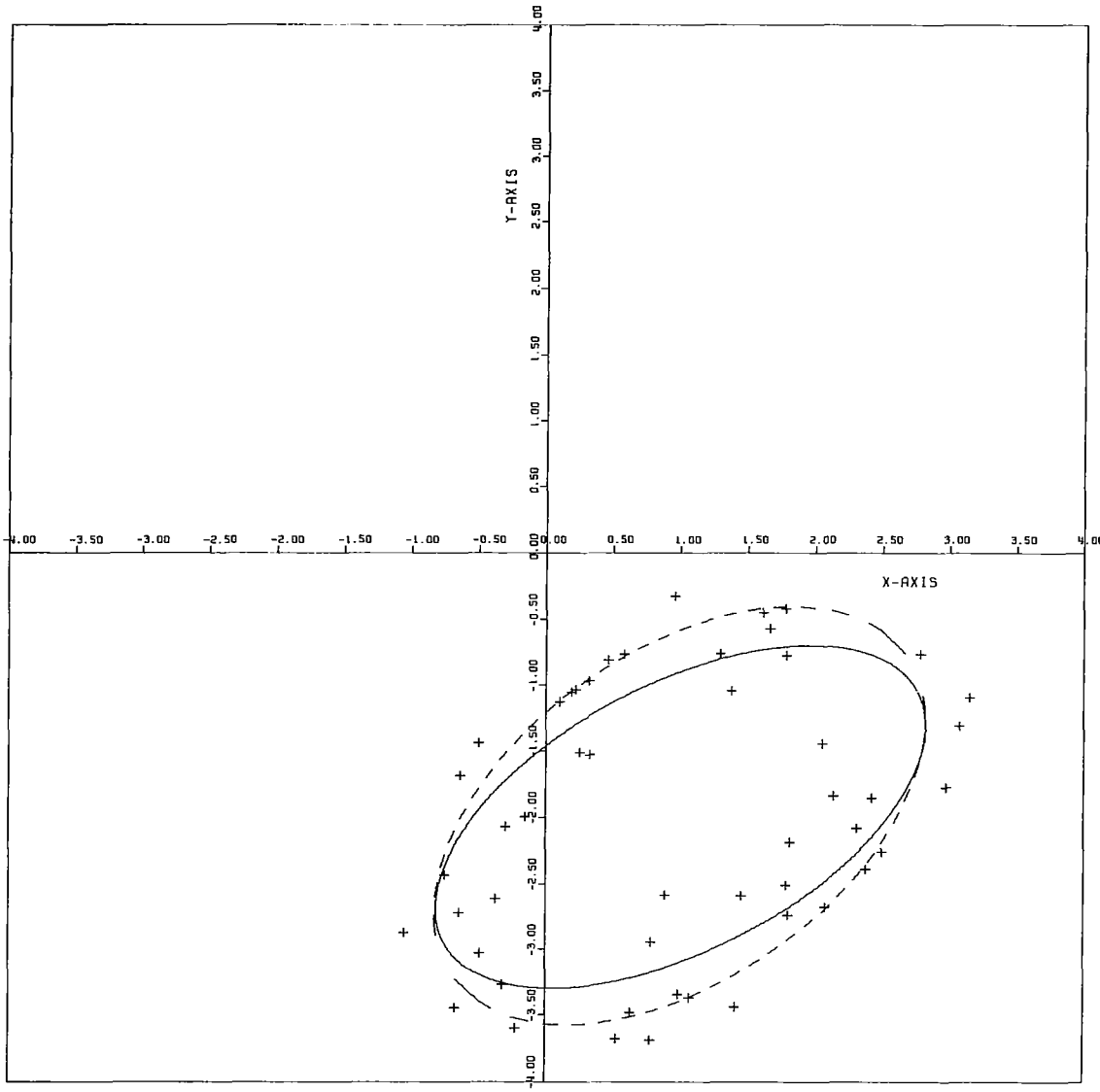
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 1.765
Y-RADIUS = 1.266
X-TRANSLATION = 1.132
Y-TRANSLATION = -1.817
ROTATION IN DEGREES = 35.374

Fig. 11--Continued .

FIELD OF VIEW



50 DATA POINTS
STANDARD DEVIATION = 0.3

REFERENCE ELLIPSE

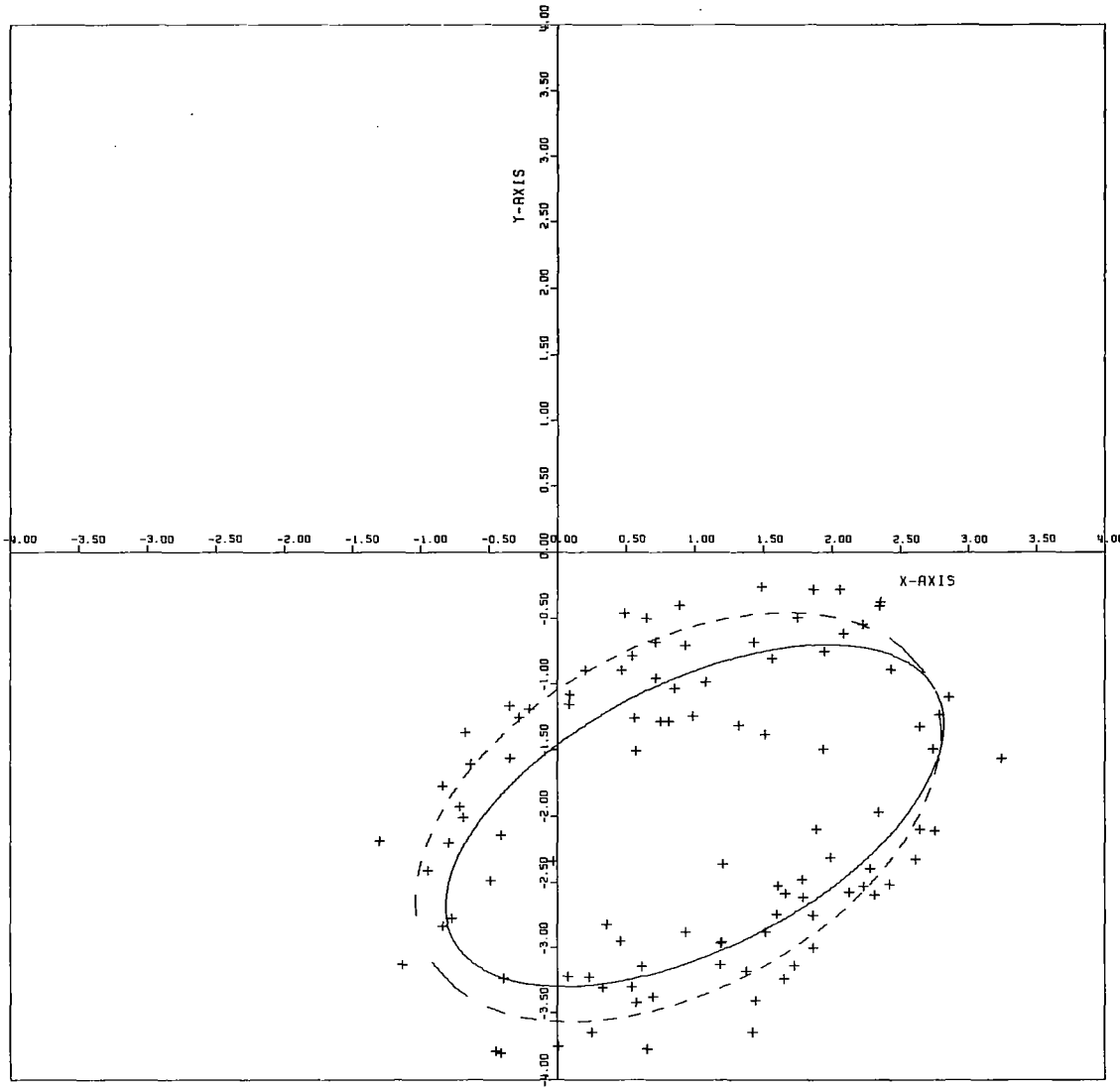
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.078
Y-RADIUS = 1.234
X-TRANSLATION = 0.994
Y-TRANSLATION = -1.991
ROTATION IN DEGREES = 36.561

Fig. 11--Continued .

FIELD OF VIEW



100 DATA POINTS
STANDARD DEVIATION = 0.3

REFERENCE ELLIPSE

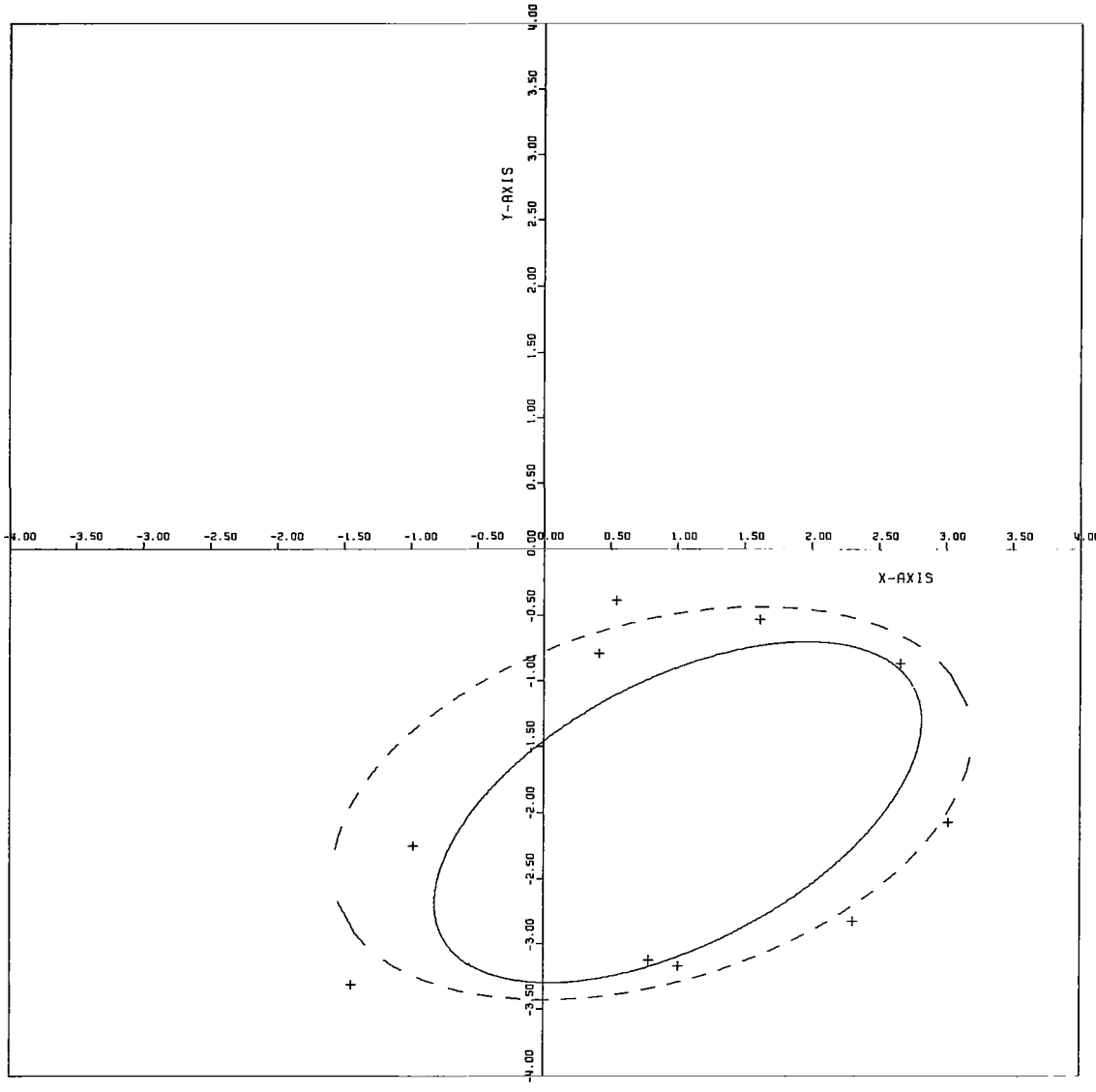
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.109
Y-RADIUS = 1.289
X-TRANSLATION = 0.881
Y-TRANSLATION = -2.012
ROTATION IN DEGREES = 31.452

Fig. 11--Concluded .

FIELD OF VIEW



10 DATA POINTS
STANDARD DEVIATION = 0.4

REFERENCE ELLIPSE

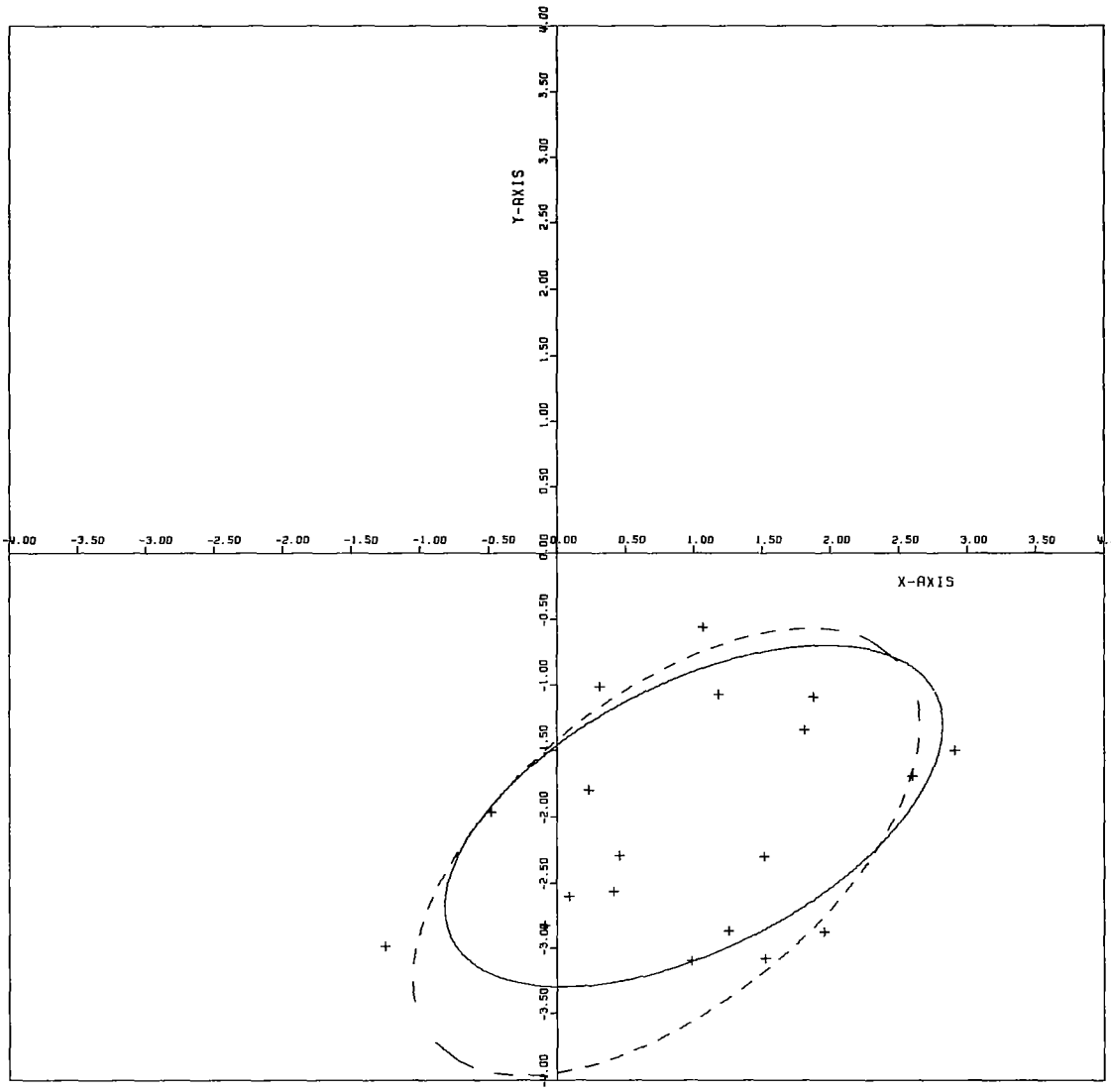
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.467
Y-RADIUS = 1.362
X-TRANSLATION = 0.807
Y-TRANSLATION = -1.931
ROTATION IN DEGREES = 17.597

Fig. 12--Ellipses fitted to data points, $\sigma = 0.4$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.4

REFERENCE ELLIPSE

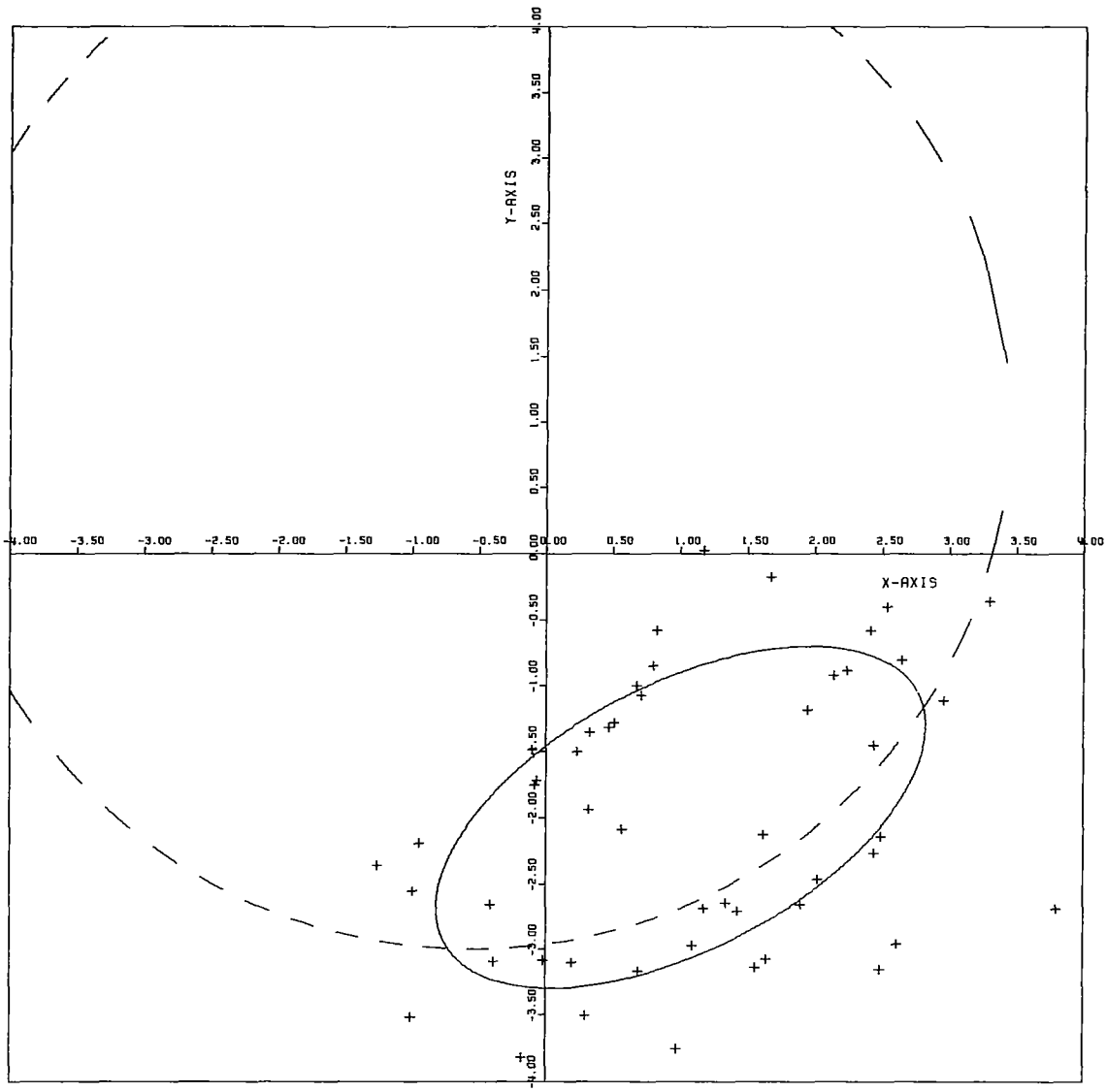
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 2.230
Y-RADIUS = 1.160
X-TRANSLATION = 0.799
Y-TRANSLATION = -2.268
ROTATION IN DEGREES = 40.669

Fig. 12--Continued .

FIELD OF VIEW



50 DATA POINTS
STANDARD DEVIATION = 0.4

REFERENCE ELLIPSE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 4.000
Y-RADIUS = 4.000
X-TRANSLATION = -0.558
Y-TRANSLATION = 1.001
ROTATION IN DEGREES = 6.505

Fig. 12--Continued .

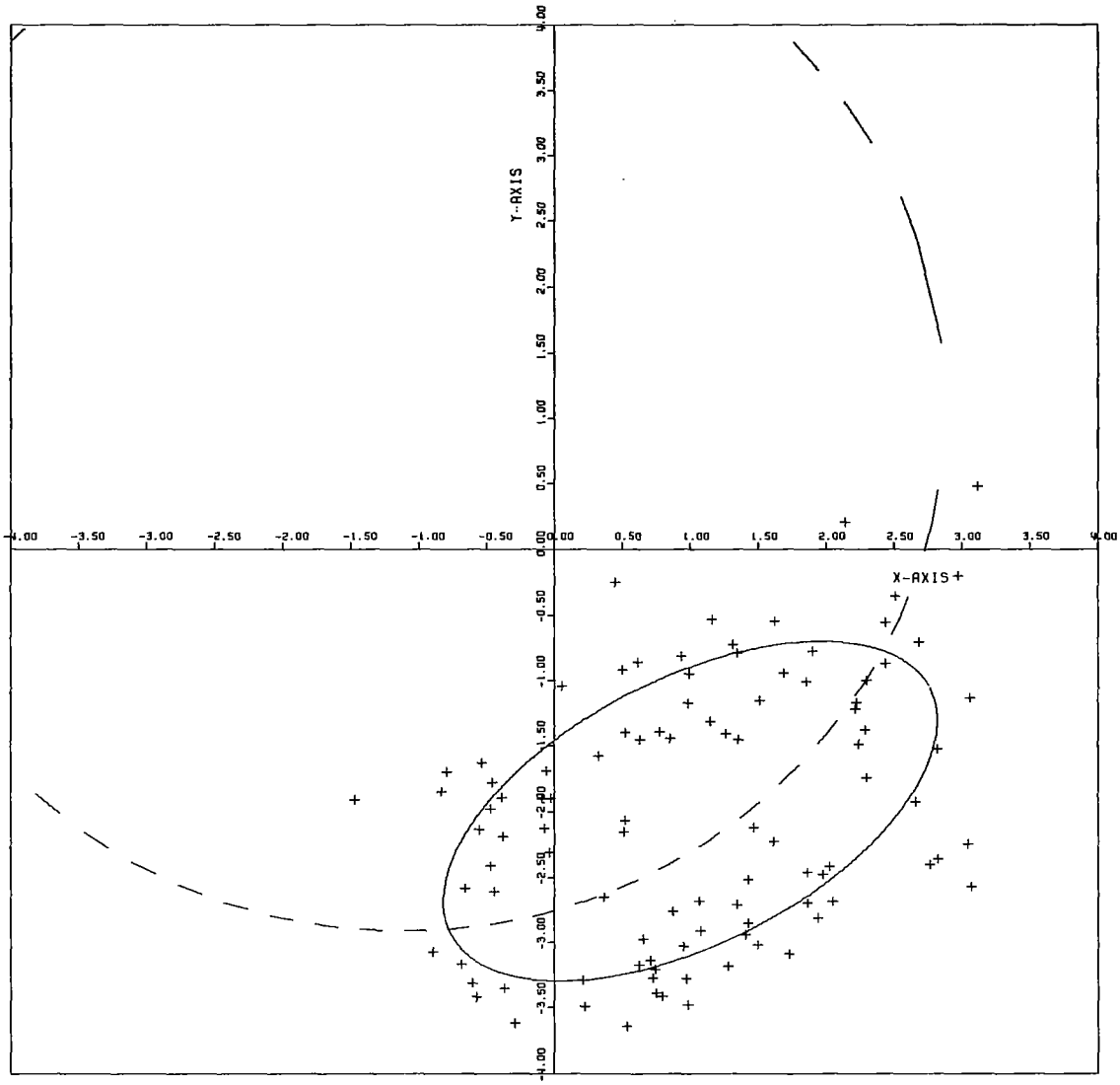
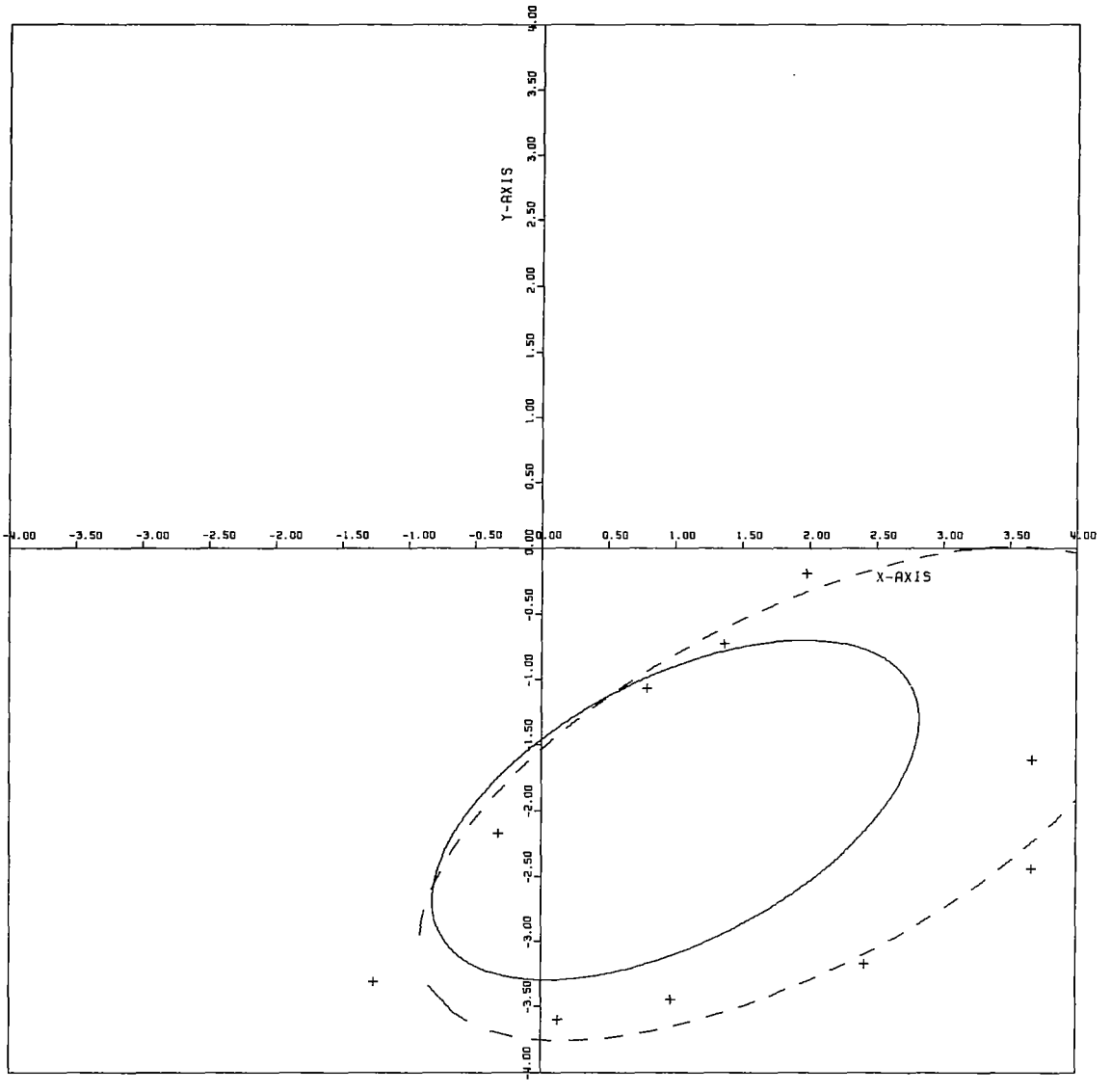


Fig. 12--Concluded .

FIELD OF VIEW



10 DATA POINTS
STANDARD DEVIATION = 0.5

REFERENCE ELLIPSE

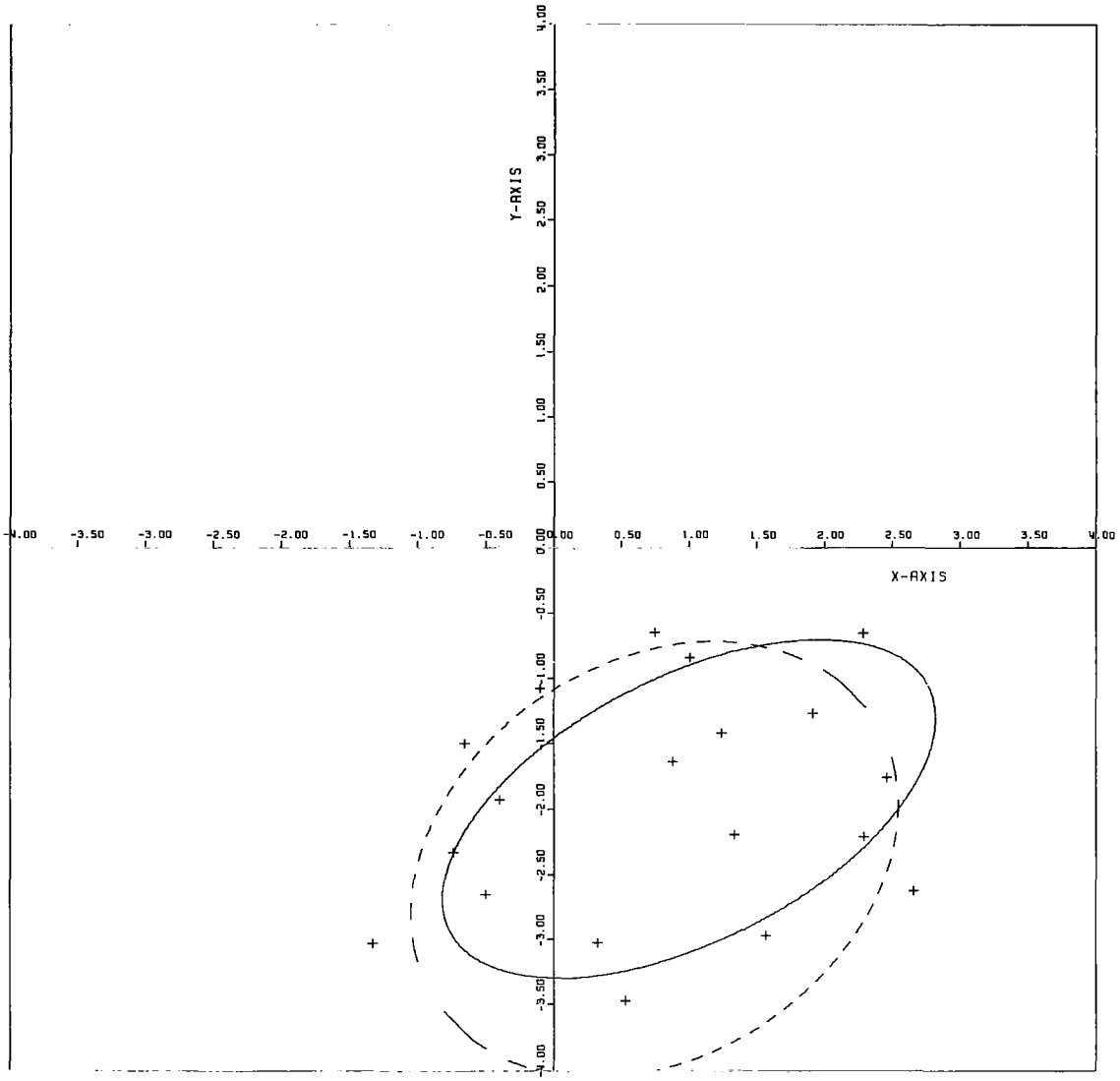
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 3.070
Y-RADIUS = 1.344
X-TRANSLATION = 1.851
Y-TRANSLATION = -1.874
ROTATION IN DEGREES = 28.623

Fig. 13--Ellipses fitted to data points, $\sigma = 0.5$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.5

REFERENCE ELLIPSE

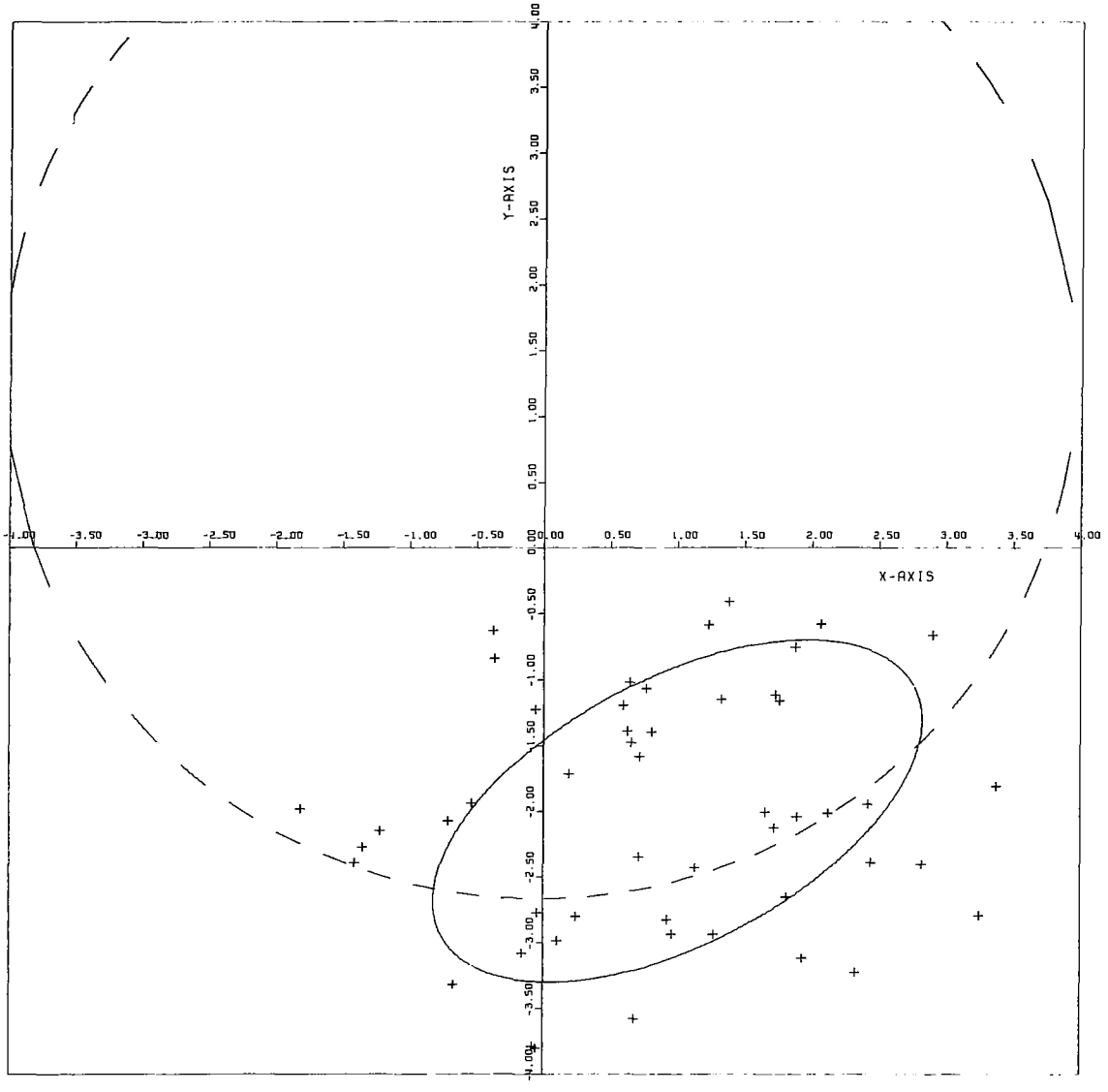
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 1.947
Y-RADIUS = 1.491
X-TRANSLATION = 0.747
Y-TRANSLATION = -2.386
ROTATION IN DEGREES = 36.845

Fig. 13--Continued .

FIELD OF VIEW



50 DATA POINTS
STANDARD DEVIATION = 0.5

REFERENCE ELLIPSE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES ELLIPSE

X-RADIUS = 4.000
Y-RADIUS = 4.000
X-TRANSLATION = -0.042
Y-TRANSLATION = 1.333
ROTATION IN DEGREES = 7.679

Fig. 13--Continued .

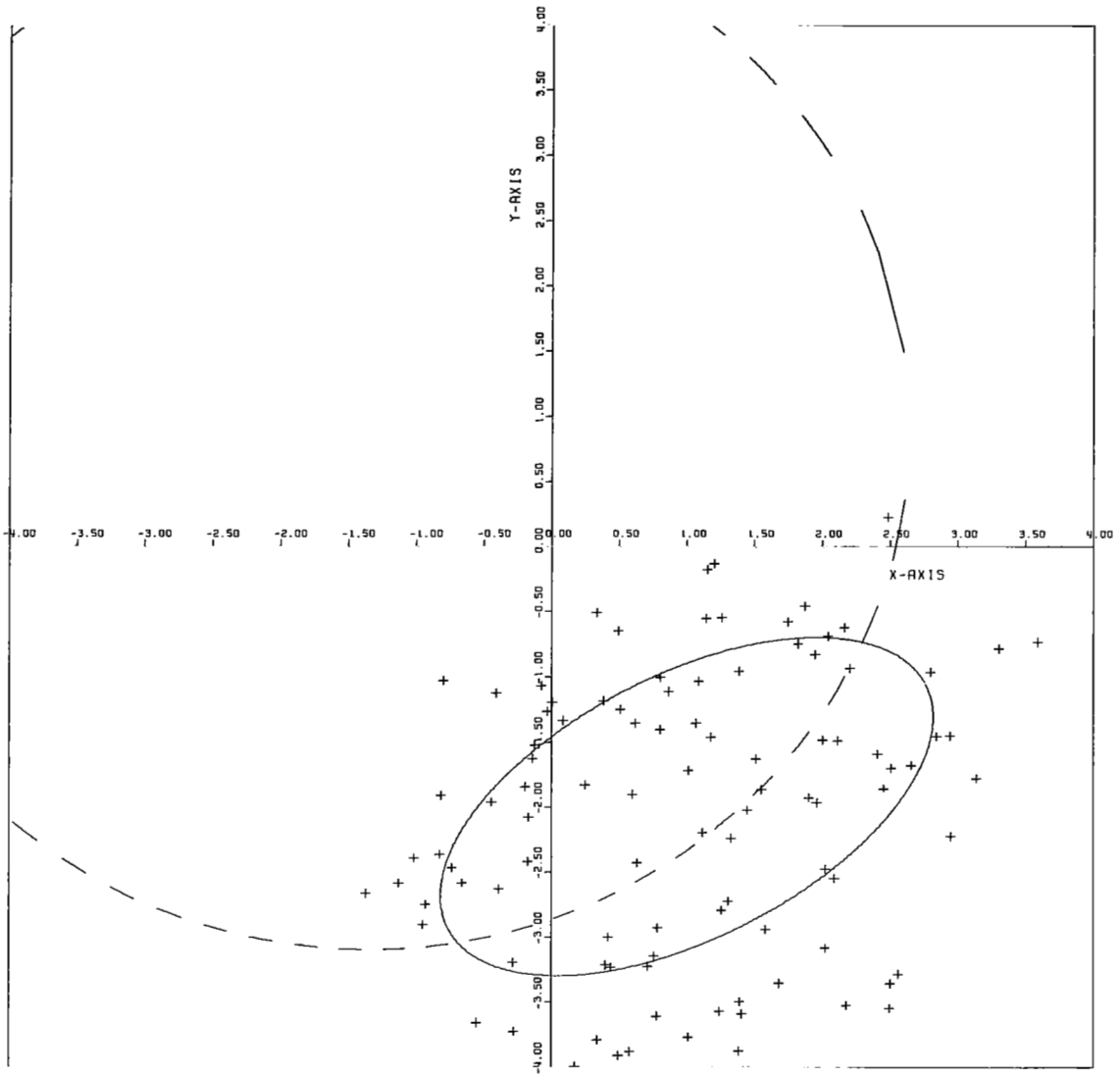


Fig. 13--Concluded .

CHAPTER V

THE RECOGNITION OF RECTANGULAR PLANAR PATTERNS

5.1 Introduction

The objective of this chapter is to investigate the feasibility of employing either or both of the estimation schemes which are discussed in Chapter III to recognize rectangular planar patterns. Again, by the term "recognition" is meant the estimation of the five parameters which characterize a rectangle having arbitrary size and shape, as well as arbitrary position (translation and rotation) in a planar field of view.

In Section 5.2 the statement of the problem is formulated and the recognition strategy which will be investigated is discussed. The results which are obtained from the two minimization schemes for rectangles having known parameter vectors (no noise) are then analyzed in Section 5.3.

The iterative minimization scheme is employed in Section 5.4 to estimate the parameters associated with rectangles whose boundary points are corrupted with noise, and the results using this scheme are discussed. Finally, Section 5.5 contains a brief summary of the results and conclusions which have been reached in this chapter.

5.2 Statement of Problem

In order to further test the recognition techniques developed for elliptical objects, a second class of patterns was considered. Rectangular patterns were chosen for this purpose because they are simple geometric patterns and yet do not possess a simple analytic representation. Furthermore, a rectangle has several properties in common with an ellipse. Both of these patterns are convex, and both are symmetrical about two orthogonal axes. Thus, a rectangle may be characterized by a set of five parameters in a manner quite similar to an ellipse. Figure 14 shows a rectangle which has been translated and rotated with respect to the x-y coordinate system. This rectangle is characterized by its w-axis radius R_w and z-axis radius R_z, and by the x- and y-translation of its center (A' and B', respectively), as well as by its rotation θ'.

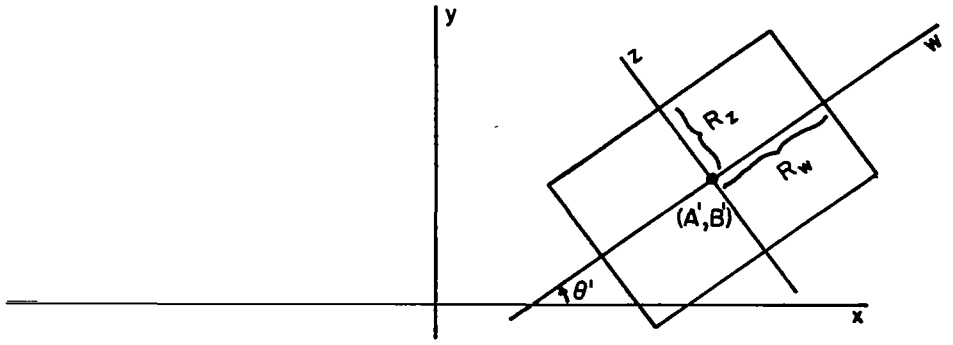


Fig. 14--Parameters of a rectangle in the x, y-reference frame.

Therefore, the parameter vector characterizing a rectangle may be expressed as

$$\vec{c}' = \begin{bmatrix} R_w \\ R_z \\ A' \\ B' \\ \theta' \end{bmatrix} \qquad (5\text{-}1)$$

The strategy which was used to recognize rectangular patterns was to present a number of different size and shape rectangles to both the one step minimization method and the iterative minimization scheme, and to determine what relationship, if any, existed between the parameter vectors of the fitted ellipses and the parameter vectors of the corresponding rectangles. This strategy is motivated by the fact that the noiseless data points which lie on the boundary of a rectangle may be considered as being noisy data points which originally belonged on the boundary of some ellipse. If a relationship can be found between the parameter vectors of the fitted ellipses and the parameter vectors of the corresponding rectangles, then it will be possible to compute the parameter vector of an unknown rectangle after the parameter vector of its associated fitted ellipse is determined.

5.3 Parameter Estimation for Noise-Free Rectangular Patterns

The results of using the one step minimization method to recognize a rectangle are shown in Tables 3, 4, 5, and 6. Each table corresponds to a different number of data points on the boundary of the rectangle,

the number of data points being 8, 20, 48, or 100, respectively. These data points were generated by the DATA subroutine which is described in Appendix I.

A total of twenty rectangles were to be recognized. Ten of the rectangles have a w-axis radius equal to one unit of length, with the z-axis radius varying from 0.1 to 1.0 units of length in increments of 0.1. The other ten rectangles are exactly twice the dimensions of the first ten. The translation and rotation parameters of the rectangles were chosen to be the same as those which were used for the ellipse which was discussed in Chapter IV. Thus

$$\begin{aligned} c_3' &= \text{x-translation} = A' = 1.0 \\ c_4' &= \text{y-translation} = B' = -2.0 \\ c_5' &= \text{rotation in radians} = \theta' = 0.5 \end{aligned} \qquad (5.2)$$

In the tables the radii of the rectangles are denoted by R while the radii of the fitted ellipses are denoted by r. The radii of the fitted ellipses are, of course, related to the first two components of their characterizing parameter vector, $\vec{c}$, by Eqn. (5.3).

$$r_w = \sqrt{\frac{1}{c_1}}$$

$$r_z = \sqrt{\frac{1}{c_2}} \qquad (5.3)$$

Inspection of Tables 3, 4, 5, and 6 indicates that the one step minimization method is not very effective in recognizing the rectangles. For the most part the translation and rotation parameters of the fitted ellipses are quite close in value to the corresponding parameters of the given rectangles. However, there is no recognizable correspondence between the w-axis and z-axis radii of the fitted ellipses and the respective radii of the rectangles.

A remark should be made at this point regarding the results which one would intuitively expect to obtain. First of all, since an ellipse and rectangle have similar geometric properties (convexity and symmetry about two orthogonal axes) one would expect that the known rectangles and the corresponding fitted ellipses would have identical coordinates for their centers, as well as identical rotation angles. On the other hand, it is difficult to predict the exact relation between each radius of a known rectangle and the corresponding radius of the fitted ellipse. However, again due to symmetry, one would expect that the ratio of the radii of a fitted ellipse would be nearly equal to the ratio of the radii of the corresponding known rectangle, being more or less independent of the rec-

tangle's size. The one step minimization method did not possess this property, however. If a recognition scheme does have this property, then the constant of proportionality relating the size of the rectangle to the size of the corresponding fitted ellipse may be determined experimentally.

Another weakness of the one step minimization method is that it is unable to fit any ellipse to some of the given rectangles. This might be expected, however, since the one step minimization was unable to fit an ellipse to an ellipse under high noise conditions, as was pointed out in Chapter IV. Thus, the one step minimization method does not appear to be a good method for estimating the parameters of a rectangle.

The iterative minimization scheme was next employed to estimate the parameters of these same rectangles. The same values were used for the range constraints as were used in the recognition of ellipses in Chapter IV. Tables 7, 8, 9, and 10 show the results of using this scheme. The initial estimate used for the parameter vector of the fitted ellipse for the ten larger rectangles was

$$\vec{c}_e = \begin{bmatrix} 0.25 \\ 1.00 \\ 0.7 \\ -2.5 \\ 0.8 \end{bmatrix} \tag{5.4}$$

Inspection of Tables 7, 8, 9, and 10 indicates that the iterative minimization scheme is quite effective in estimating the parameters of a rectangle. It is seen that the ratio r_z/r_w corresponding to the radii of the fitted ellipse is equal (to within three decimal places) to the radii R_z/R_w of the rectangle which is to be recognized. Thus the ratio of the radii of the fitted ellipses gives a direct indication of the shape of the rectangles to which they are fitted.

Tables 7, 8, 9, and 10 also indicate that the size of the rectangles may be determined to a reasonable degree of accuracy. When eight data points on the boundary of the rectangle are given, Table 7 shows that $R_w = 0.775\ r_w$, meaning that the rectangles' radii are 0.775 times the length of the fitted ellipses' radii. Likewise, Tables 8, 9, and 10 show that the scale factor, R_w/r_w is equal to 0.830, 0.842, and 0.845 for 20, 48, and 100 data points on the boundary of the rectangle, respectively.

One can see that the scale factor does not change appreciably when more than 48 data points are given. If one assumes that the scale

factor associated with 100 data points is essentially the same as the scale factor associated with an infinite number of data points (which seems reasonable in light of the above results) then it is possible to compare the scale factor associated with a finite number of data points with the scale factor associated with a continuous representation of the rectangle. For eight data points this ratio is 0.775/0.845 = 0.917, which means that the estimated size of the rectangle is only 91.7% of the size of the actual rectangle, although it has exactly the same shape as the actual rectangle. For 20 data points this ratio increases to 0.830/0.845 = 0.983 and for 48 data points the ratio is 0.842/0.845 = 0.997. Thus, if the rectangle is represented by twenty or more data points on its boundary, then one need merely multiply the radii of the fitted ellipse by the factor 0.845 to obtain the radii of the corresponding rectangle, having assurance that this rectangle will be at least within 2% of the size of the actual rectangle.

Tables 7, 8, 9, and 10 show another very desirable property of the iterative minimization scheme, namely, the translation and rotation parameters of the fitted ellipses have exactly the same values (to within three decimal places) as the corresponding parameters of the given rectangles. Therefore, only the first two components of the fitted ellipse's parameter vector need to be transformed in order to obtain the parameter vector of the rectangle, and this transformation is a simple scale change.

Thus, the desired relationship between the parameter vector of the fitted ellipse and the parameter vector of the associated rectangle has now been determined. If the final estimate for the parameter vector of the fitted ellipse is given by

$$\vec{c}_e = \begin{bmatrix} e_1 \\ e_2 \\ A \\ B \\ \theta \end{bmatrix} \tag{5.5}$$

then the estimate for the parameter vector of the rectangle which is to be recognized is

$$\vec{c}'_e = \begin{bmatrix} R_w \\ R_z \\ A' \\ B' \\ \theta' \end{bmatrix} = \begin{bmatrix} k/\sqrt{e_1} \\ k/\sqrt{e_2} \\ A \\ B \\ \theta \end{bmatrix} \tag{5.6}$$

where k, the scale factor, is a function of the number of data points. If the number of data points is 20 or greater, k may be taken to be 0.845.

5.4 Parameter Estimation for Noisy Rectangular Patterns

Since the iterative minimization scheme was able to effectively recognize rectangles, the question naturally arises as to how well it can recognize rectangles which are represented by noisy data points. In order to determine this a total of twenty-four different cases were considered, as was done with the ellipse in Chapter IV.

The noiseless rectangle, whose parameters are to be estimated, is characterized by the following parameter vector.

$$\vec{c}_0' = \begin{bmatrix} R_w \\ R_z \\ A' \\ B' \\ \theta' \end{bmatrix} = \begin{bmatrix} 2.00 \\ 1.00 \\ 1.00 \\ -2.00 \\ 0.50 \end{bmatrix} \qquad (5.7)$$

The last three components have the same value as they did for the ellipse considered in Chapter IV. The noise levels are also the same as they were previously, namely, $\sigma = 0.0$, 0.1, 0.2, 0.3, 0.4, and 0.5. Also, all of the parameters associated with the iterative minimization scheme were given the same values as they had in Chapter IV, with the exception of the initial parameter vector estimate $\vec{c}_e$. It is

$$\vec{c}_e = \begin{bmatrix} e_1 \\ e_2 \\ A \\ B \\ \theta \end{bmatrix} = \begin{bmatrix} 0.25 \\ 1.00 \\ 0.70 \\ -2.50 \\ 0.80 \end{bmatrix} \qquad (5.8)$$

Figures 15, 16, 17, 18, 19, and 20 show the results of estimating the parameter vector of a rectangle using the iterative minimization scheme. Before discussing these results it should be pointed out that the radii of the rectangle were computed by using the scale factor associated with the appropriate number of data points. Thus, for example, for the six cases in which the rectangle was represented by 20 data points the scale factor which was used was 0.830, and not 0.845. By doing this, any error in the estimated parameter vector is due to the noisy data points.

An examination of Figure 2 reveals that the x-radii and y-radii (w-axis radii and x-axis radii, respectively) of the fitted rectangles

differ in the third decimal place from the corresponding radii of the reference rectangles. This is due to rounding off the scale factors to the third decimal place. This error is entirely negligible compared to the error resulting from the noisy data points.

In some of the figures there are not as many noisy data points as the number which is indicated. This is due to the fact that some of the noisy data points fell outside of the field of view. As before, these data points are considered to be valid points for the iterative minimization scheme to use, being analogous to measurement noise.

An inspection of Figures 15, 16, 17, 18, 19, and 20 shows that fitting an ellipse to a set of data points belonging on the boundary of a rectangle is an effective method by which to estimate the parameters of the rectangle when the noise level is within reasonable limits (σ = 0.0, 0.1, and 0.2).

Referring to Figure 16 (σ = 0.1) one sees that the rectangles which correspond to the fitted ellipses (the dashed line rectangles) resemble the reference rectangles very closely except in the eight data point case. It would seem that eight data points, when corrupted by noise, simply are too sparse in number for the iterative minimization scheme to yield a good estimate for the reference rectangle's parameter vector. However, it should be noted that the error in estimating the rotation angle for the eight data point case is quite acceptable, being approximately two percent. In the other three cases this error is approximately one percent or less.

For σ = 0.2 it can be seen in Figure 17 that the parameter vector estimates are beginning to deteriorate, but for the 20, 48 and 100 data point cases these estimates are still acceptable by most standards. In particular, it is seen that for these three cases the error in the rotation angle estimate is no greater than approximately four percent, which is rather small considering the scatter of the data points.

When the noise level reaches σ = 0.3, the overall effectiveness of the iterative minimization scheme becomes questionable. The fitted rectangles have a tendency to be larger than the reference rectangles. However, one good point which can be made is that the estimate for the rotation angle in all four cases does not exceed four percent, which means that this estimate has not been affected to any extent by the increase in noise level from σ = 0.2 to σ = 0.3.

Figures 19 and 20 indicate that for high noise levels ($\sigma = 0.4$ and 0.5) the recognition capability of the iterative minimization scheme has completely deteriorated. Some improvement could be achieved for the cases in which the fitted rectangles have radii equal to their constraint value. In these cases the fitted rectangle is a square which concentrates the data points in one of its corners, with approximately one half of the data points on the inside of the square and one half on the outside. This situation is very similar to that which occurred in the recognition of ellipses under high noise conditions, and it can be remedied in exactly the same manner as described in Chapter IV.

5.5 Summary

This chapter investigated the feasibility of utilizing either the one step minimization method or the iterative minimization scheme to estimate the parameters of a rectangle when noisefree data points lying on the rectangle's boundary are given. If the parameters are estimated with sufficient precision then the rectangle has been "recognized" correctly.

It was found that the one step minimization method was completely inadequate in its capability to estimate the parameters of given rectangles. While it did do a reasonable job in estimating the translation and rotation parameters, the two major shortcomings of this method were

(1) the ellipse which was fitted to the data points did not have the same shape as the given rectangle, i.e., the ratio of the radii of the fitted ellipse was not identical to the corresponding ratio of the radii of the given rectangle,

and

(2) the size of the fitted ellipse did not double when the size of the given rectangle doubled.

The iterative minimization scheme, on the other hand, did not have these shortcomings. Not only did it estimate the translation and rotation parameters very precisely, but the ellipse which it fitted to the data points had radii whose ratio was identical to that of the given rectangle, and this ratio was independent of the size of the given rectangle. It was therefore possible to experimentally determine a scale factor relating the size of the fitted ellipse's radii to the radii of the given rectangle.

Since the iterative minimization scheme had the capability to precisely estimate the parameters of a rectangle whose boundary points were noise free, the next step was to determine the degradation in the parameter vector estimates in situations for which the data points were noisy. Reference to Figures 15, 16, and 17 indicates that for moderate

levels of noise (σ = 0.0, 0.1, and 0.2) the iterative minimization scheme did a very satisfactory job of recognizing the rectangles. Special notice should be taken concerning the accuracy with which the rotation angle was estimated. Excluding the eight data point case, this error was never greater than four percent for these moderate noise levels.

The recognition scheme produced results of questionable value for noise level σ = 0.3. Although the rotation angle was still estimated with good precision (maximum of four percent error), the size of the fitted rectangle tended to be larger than the size of the reference rectangle. It can be said that σ = 0.3 represents the maximum noise level for which the iterative minimization scheme produces useful results for the particular set of rectangles investigated.

For larger noise levels (σ = 0.4 and 0.5) the iterative minimization scheme was not able to do a satisfactory job of recognizing the rectangles at all. This is not at all surprising, since even a human being would have difficulty trying to fit a rectangle to the data as shown on Figures 19 and 20.

TABLE 3: PARAMETERS OF ELLIPSES FITTED TO RECTANGLES BY THE ONE STEP MINIMIZATION METHOD (8 DATA POINTS)

R_w	R_z	R_z/R_w	r_w	r_z	r_z/r_w	A	B	θ
1.0	0.1	0.100	1.331	0.077	.058	1.152	-1.917	.496
1.0	0.2	0.200	1.768	0.202	.114	.962	-2.023	.499
1.0	0.3	0.300	1.755	0.298	.170	.996	-2.007	.499
1.0	0.4	0.400	1.773	0.396	.223	1.003	-2.008	.499
1.0	0.5	0.500	1.799	0.491	.273	1.006	-2.016	.498
1.0	0.6	0.600	1.834	0.585	.319	1.007	-2.018	.497
1.0	0.7	0.700	1.886	0.675	.358	1.010	-2.025	.496
1.0	0.8	0.800	1.957	0.761	.389	1.014	-2.034	.495
1.0	0.9	0.900	2.063	0.842	.408	1.017	-2.045	.493
1.0	1.0	1.000	2.234	0.915	.410	1.020	-2.060	.491
2.0	0.2	0.100	3.517	0.201	.057	.995	-2.005	.500
2.0	0.4	0.200	3.540	0.395	.112	1.003	-2.007	.500
2.0	0.6	0.300	3.670	0.584	.160	1.008	-2.017	.499
2.0	0.8	0.400	3.918	0.761	.194	1.014	-2.034	.499
2.0	1.0	0.500	4.481	0.915	.204	1.021	-2.060	.498
2.0	1.2	0.600	6.801	1.030	.151	1.010	-2.109	.497
2.0	1.4	0.700	*					
2.0	1.6	0.800	*					
2.0	1.8	0.900	*					
2.0	2.0	1.000	*					

*Not able to fit ellipse data.

Note: R corresponds to reference rectangles' radii

r corresponds to fitted ellipses' radii

All reference rectangles have A = 1.0, B = -2.0, and θ = 0.5

TABLE 4: PARAMETERS OF ELLIPSES FITTED TO RECTANGLES BY THE ONE STEP MINIMIZATION METHOD (20 DATA POINTS)

R_w	R_z	R_z/R_w	r_w	r_z	r_z/r_w	A	B	θ
1.0	0.1	0.100	0.699	0.056	0.080	1.238	-1.865	0.498
1.0	0.2	0.200	1.326	0.209	0.158	0.988	-2.007	0.500
1.0	0.3	0.300	1.330	0.314	0.236	1.003	-2.002	0.500
1.0	0.4	0.400	1.335	0.418	0.313	1.003	-2.004	0.499
1.0	0.5	0.500	1.340	0.520	0.388	1.004	-2.008	0.498
1.0	0.6	0.600	1.347	0.622	0.462	1.005	-2.012	0.497
1.0	0.7	0.700	1.354	0.722	0.533	1.008	-2.016	0.496
1.0	0.8	0.800	1.364	0.819	0.600	1.011	-2.022	0.494
1.0	0.9	0.900	1.377	0.914	0.664	1.014	-2.029	0.491
1.0	1.0	1.000	1.394	1.005	0.721	1.018	-2.038	0.487
2.0	0.2	0.100	2.643	0.209	0.079	1.017	-1.992	0.500
2.0	0.4	0.200	2.671	0.418	0.156	1.002	-2.005	0.500
2.0	0.6	0.300	2.695	0.622	0.231	1.005	-2.012	0.499
2.0	0.8	0.400	2.729	0.819	0.300	1.011	-2.022	0.499
2.0	1.0	0.500	2.789	1.004	0.360	1.018	-2.038	0.498
2.0	1.2	0.600	2.902	1.168	0.402	1.029	-2.063	0.497
2.0	1.4	0.700	3.165	1.289	0.407	1.046	-2.104	0.495
2.0	1.6	0.800	4.429	1.309	0.296	1.062	-2.187	0.492
2.0	1.8	0.900	*					
2.0	2.0	1.000	*					

*Not able to fit ellipse to data.

TABLE 5: PARAMETERS OF ELLIPSES FITTED TO RECTANGLES BY THE ONE STEP MINIMIZATION METHOD (48 DATA POINTS)

R_w	R_z	R_z/R_w	r_w	r_z	r_z/r_w	A	B	θ
1.0	0.1	0.100	1.307	0.111	0.085	0.640	-2.206	0.515
1.0	0.2	0.200	1.246	0.207	0.166	1.001	-2.001	0.503
1.0	0.3	0.300	1.277	0.318	0.249	1.001	-2.002	0.501
1.0	0.4	0.400	1.282	0.424	0.331	1.001	-2.005	0.500
1.0	0.5	0.500	1.284	0.529	0.412	1.003	-2.007	0.499
1.0	0.6	0.600	1.288	0.632	0.491	1.004	-2.010	0.498
1.0	0.7	0.700	1.292	0.735	0.569	1.006	-2.013	0.497
1.0	0.8	0.800	1.299	0.835	0.643	1.009	-2.018	0.495
1.0	0.9	0.900	1.307	0.933	0.714	1.012	-2.024	0.491
1.0	1.0	1.000	1.317	1.028	0.781	1.015	-2.031	0.486
2.0	0.2	0.100	2.537	0.211	0.083	0.977	-2.014	0.500
2.0	0.4	0.200	2.564	0.424	0.165	1.002	-2.004	0.500
2.0	0.6	0.300	2.576	0.633	0.246	1.004	-2.010	0.500
2.0	0.8	0.400	2.598	0.835	0.321	1.009	-2.018	0.499
2.0	1.0	0.500	2.635	1.028	0.390	1.015	-2.032	0.498
2.0	1.2	0.600	2.702	1.203	0.445	1.025	-2.052	0.497
2.0	1.4	0.700	2.845	1.343	0.472	1.040	-2.086	0.495
2.0	1.6	0.800	3.323	1.401	0.422	1.065	-2.152	0.492
2.0	1.8	0.900	*					
2.0	2.0	1.000	*					

*Not able to fit ellipse to data.

TABLE 6: PARAMETERS OF ELLIPSES FITTED TO RECTANGLES BY THE ONE STEP MINIMIZATION METHOD (100 DATA POINTS)

R_w	R_z	R_z/R_w	r_w	r_z	r_z/r_w	A	B	θ
1.0	0.1	0.100	0.491	0.044	0.090	0.722	-2.141	0.506
1.0	0.2	0.200	1.256	0.213	0.170	1.015	-1.992	0.503
1.0	0.3	0.300	1.265	0.320	0.253	0.995	-2.004	0.501
1.0	0.4	0.400	1.269	0.425	0.335	1.000	-2.004	0.500
1.0	0.5	0.500	1.273	0.530	0.416	1.002	-2.006	0.499
1.0	0.6	0.600	1.277	0.635	0.497	1.004	-2.009	0.498
1.0	0.7	0.700	1.281	0.737	0.575	1.006	-2.013	0.496
1.0	0.8	0.800	1.287	0.838	0.651	1.008	-2.017	0.495
1.0	0.9	0.900	1.295	0.937	0.724	1.011	-2.023	0.491
1.0	1.0	1.000	1.305	1.032	0.791	1.015	-2.030	0.486
2.0	0.2	0.100	2.552	0.214	0.084	0.995	-2.005	0.497
2.0	0.4	0.200	2.543	0.425	0.167	1.003	-2.004	0.499
2.0	0.6	0.300	2.556	0.634	0.248	1.005	-2.009	0.499
2.0	0.8	0.400	2.577	0.838	0.325	1.008	-2.018	0.499
2.0	1.0	0.500	2.611	1.032	0.395	1.015	-2.030	0.498
2.0	1.2	0.600	2.672	1.209	0.452	1.024	-2.050	0.497
2.0	1.4	0.700	2.801	1.353	0.483	1.039	-2.083	0.495
2.0	1.6	0.800	3.215	1.417	0.441	1.064	-2.146	0.492
2.0	1.8	0.900	*					
2.0	2.0	1.000	*					

*Not able to fit ellipse to data.

TABLE 7: PARAMETERS OF ELLIPSES FITTED TO RECTANGLES BY THE ITERATIVE MINIMIZATION METHOD (8 DATA POINTS)

R_w	R_z	R_z/R_w	r_w	r_z	r_z/r_w	A	B	θ
2.0	0.2	0.100	2.582	0.258	0.100	1.000	-2.000	0.500
2.0	0.4	0.200	2.582	0.516	0.200	1.000	-2.000	0.500
2.0	0.6	0.300	2.582	0.775	0.300	1.000	-2.000	0.500
2.0	0.8	0.400	2.582	1.033	0.400	1.000	-2.000	0.500
2.0	1.0	0.500	2.582	1.291	0.500	1.000	-2.000	0.500
2.0	1.2	0.600	2.582	1.549	0.600	1.000	-2.000	0.500
2.0	1.4	0.700	2.582	1.807	0.700	1.000	-2.000	0.500
2.0	1.6	0.800	2.582	2.066	0.800	1.000	-2.000	0.500
2.0	1.8	0.900	2.582	2.324	0.900	1.000	-2.000	0.500
2.0	2.0	1.000	2.582	2.582	1.000	1.000	-2.000	0.454

r - ellipse radii

R - rectangle radii

$$\text{Scale Factor}\Big|_{\text{8 data points}} = k_8 = \frac{R_w}{r_w} = \frac{R_z}{r_z} = 0.775$$

TABLE 8: PARAMETERS OF ELLIPSES FITTED TO RECTANGLES BY THE ITERATIVE MINIMIZATION METHOD (20 DATA POINTS)

R_w	R_z	R_z/R_w	r_w	r_z	r_z/r_w	A	B	θ
2.0	0.2	0.100	2.409	0.241	0.100	1.000	-2.000	0.500
2.0	0.4	0.200	2.409	0.482	0.200	1.000	-2.000	0.500
2.0	0.6	0.300	2.409	0.723	0.300	1.000	-2.000	0.500
2.0	0.8	0.400	2.409	0.963	0.400	1.000	-2.000	0.500
2.0	1.0	0.500	2.409	1.204	0.500	1.000	-2.000	0.500
2.0	1.2	0.600	2.409	1.445	0.600	1.000	-2.000	0.500
2.0	1.4	0.700	2.409	1.686	0.700	1.000	-2.000	0.500
2.0	1.6	0.800	2.409	1.927	0.800	1.000	-2.000	0.500
2.0	1.8	0.900	2.409	2.168	0.900	1.000	-2.000	0.500
2.0	2.0	1.000	2.409	2.409	1.000	1.000	-2.000	0.859

r - ellipse radii

R - rectangle radii

Scale Factor | 20 data points $= k_{20} = \frac{R_w}{r_w} = \frac{R_z}{r_z} = 0.830$

TABLE 9: PARAMETERS OF ELLIPSES FITTED TO RECTANGLES BY THE ITERATIVE MINIMIZATION METHOD
(48 DATA POINTS)

R_w	R_z	R_z/R_w	r_w	r_z	r_z/r_w	A	B	θ
2.0	0.2	0.100	2.374	0.237	0.100	1.000	-2.000	0.500
2.0	0.4	0.200	2.374	0.475	0.200	1.000	-2.000	0.500
2.0	0.6	0.300	2.374	0.712	0.300	1.000	-2.000	0.500
2.0	0.8	0.400	2.374	0.950	0.400	1.000	-2.000	0.500
2.0	1.0	0.500	2.374	1.187	0.500	1.000	-2.000	0.500
2.0	1.2	0.600	2.374	1.424	0.600	1.000	-2.000	0.500
2.0	1.4	0.700	2.374	1.662	0.700	1.000	-2.000	0.500
2.0	1.6	0.800	2.374	1.899	0.800	1.000	-2.000	0.500
2.0	1.8	0.900	2.374	2.137	0.900	1.000	-2.000	0.500
2.0	2.0	1.000	2.374	2.374	1.000	1.000	-2.000	0.864

r - ellipse radii

R - rectangle radii

Scale Factor | 48 data points $= k_{48} = \frac{R_w}{r_w} = \frac{R_z}{r_z} = 0.842$

TABLE 10: PARAMETERS OF ELLIPSES FITTED TO RECTANGLES BY THE ITERATIVE MINIMIZATION METHOD (100 DATA POINTS)

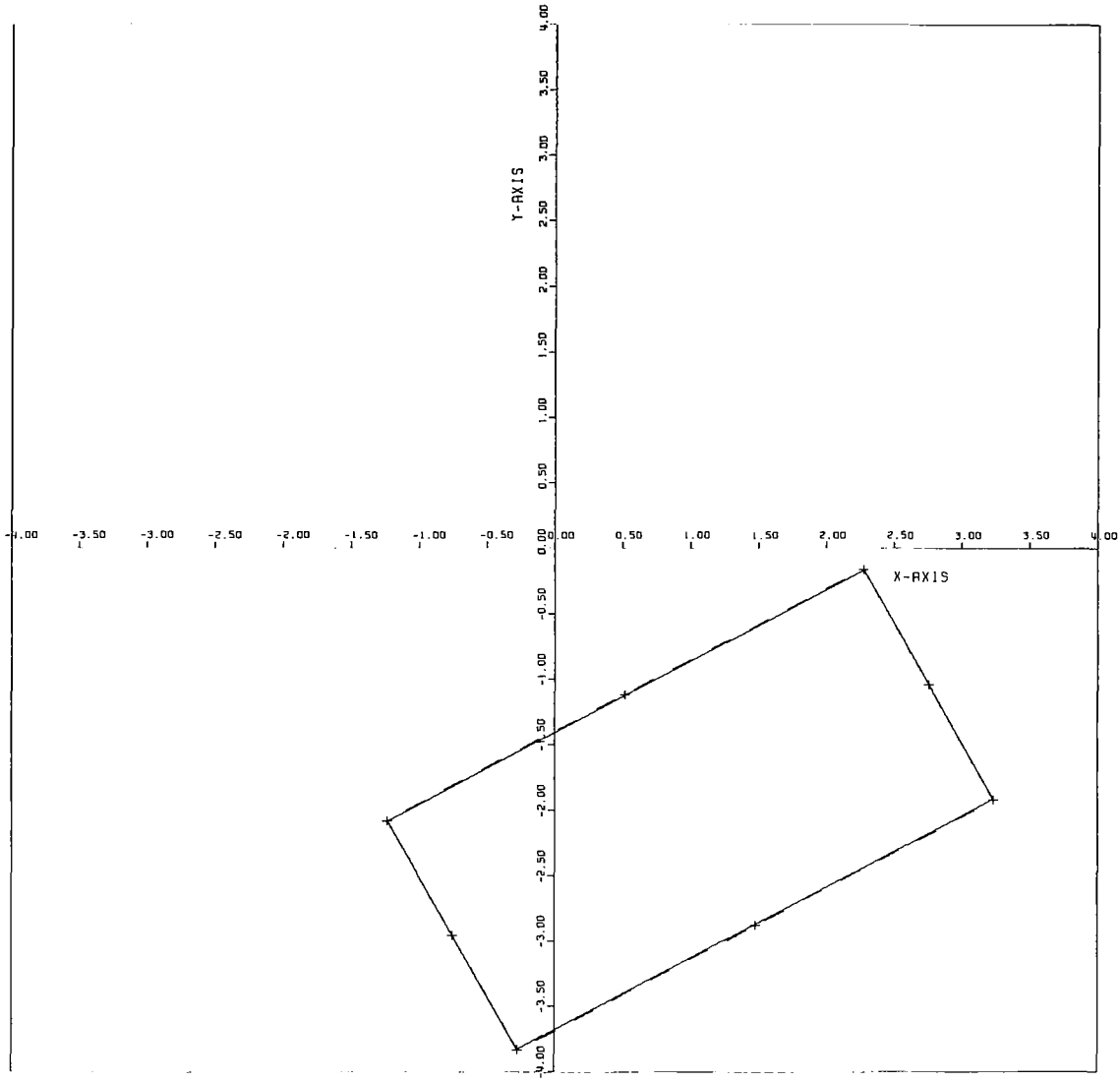
R_w	R_z	R_z/R_w	r_w	r_z	r_z/r_w	A	B	θ
2.0	0.2	0.100	2.368	0.237	0.100	1.000	-2.000	0.500
2.0	0.4	0.200	2.368	0.474	0.200	1.000	-2.000	0.500
2.0	0.6	0.300	2.368	0.710	0.300	1.000	-2.000	0.500
2.0	0.8	0.400	2.368	0.947	0.400	1.000	-2.000	0.500
2.0	1.0	0.500	2.368	1.184	0.500	1.000	-2.000	0.500
2.0	1.2	0.600	2.368	1.421	0.600	1.000	-2.000	0.500
2.0	1.4	0.700	2.368	1.658	0.700	1.000	-2.000	0.500
2.0	1.6	0.800	2.369	1.895	0.800	1.000	-2.000	0.500
2.0	1.8	0.900	2.368	2.131	0.900	1.000	-2.000	0.500
2.0	2.0	1.000	2.368	2.368	1.000	1.000	-2.000	0.500

r - ellipse radii

R - rectangle radii

Scale Factor | 100 data points $= k_{100} = \frac{R_w}{r_w} = \frac{R_z}{r_z} = 0.845$

FIELD OF VIEW



8 DATA POINTS
STANDARD DEVIATION = 0.0

REFERENCE RECTANGLE

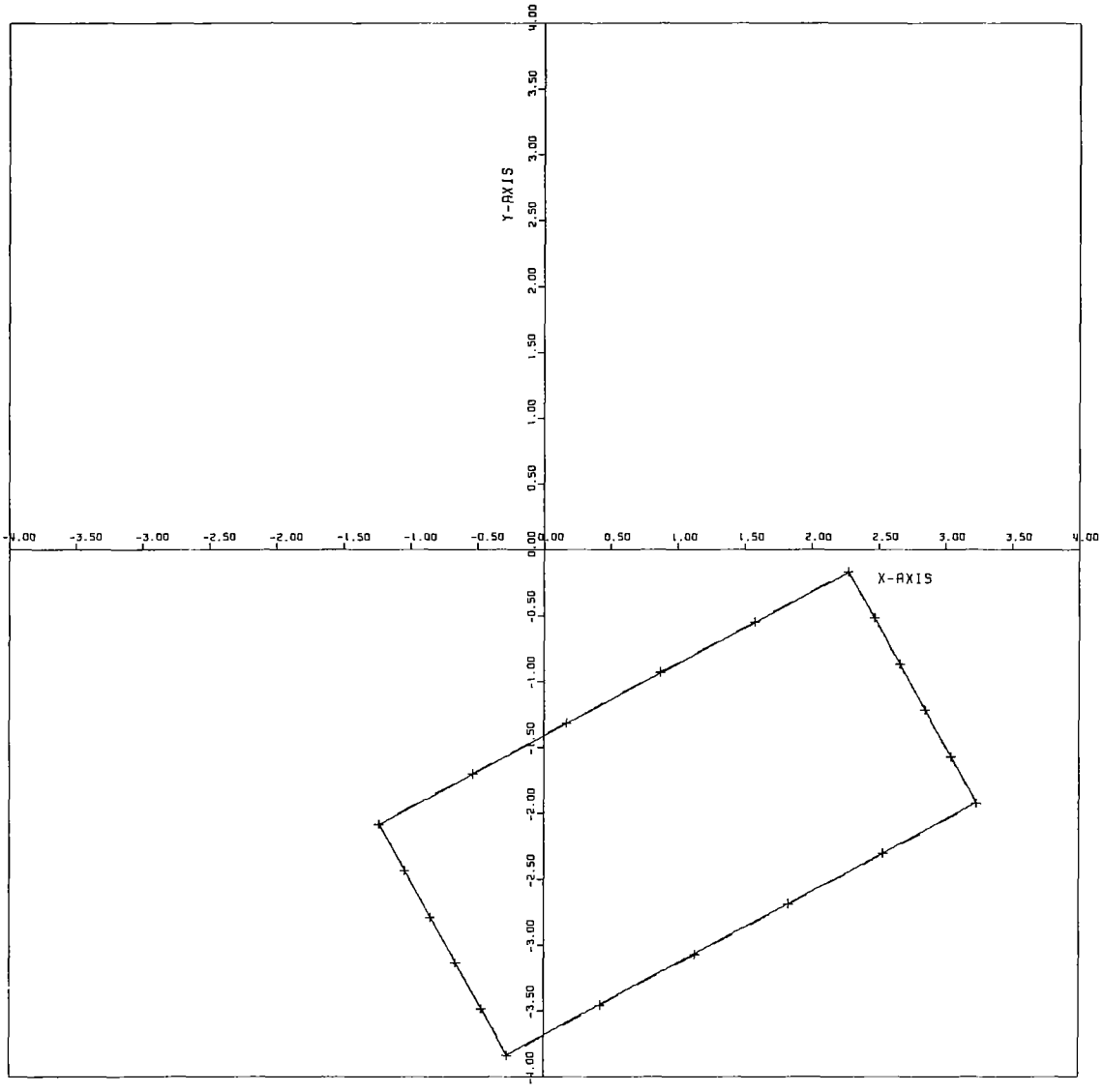
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.001
Y-RADIUS = 1.001
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

Fig. 15--Rectangles fitted to data points, $\sigma = 0.0$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.0

REFERENCE RECTANGLE

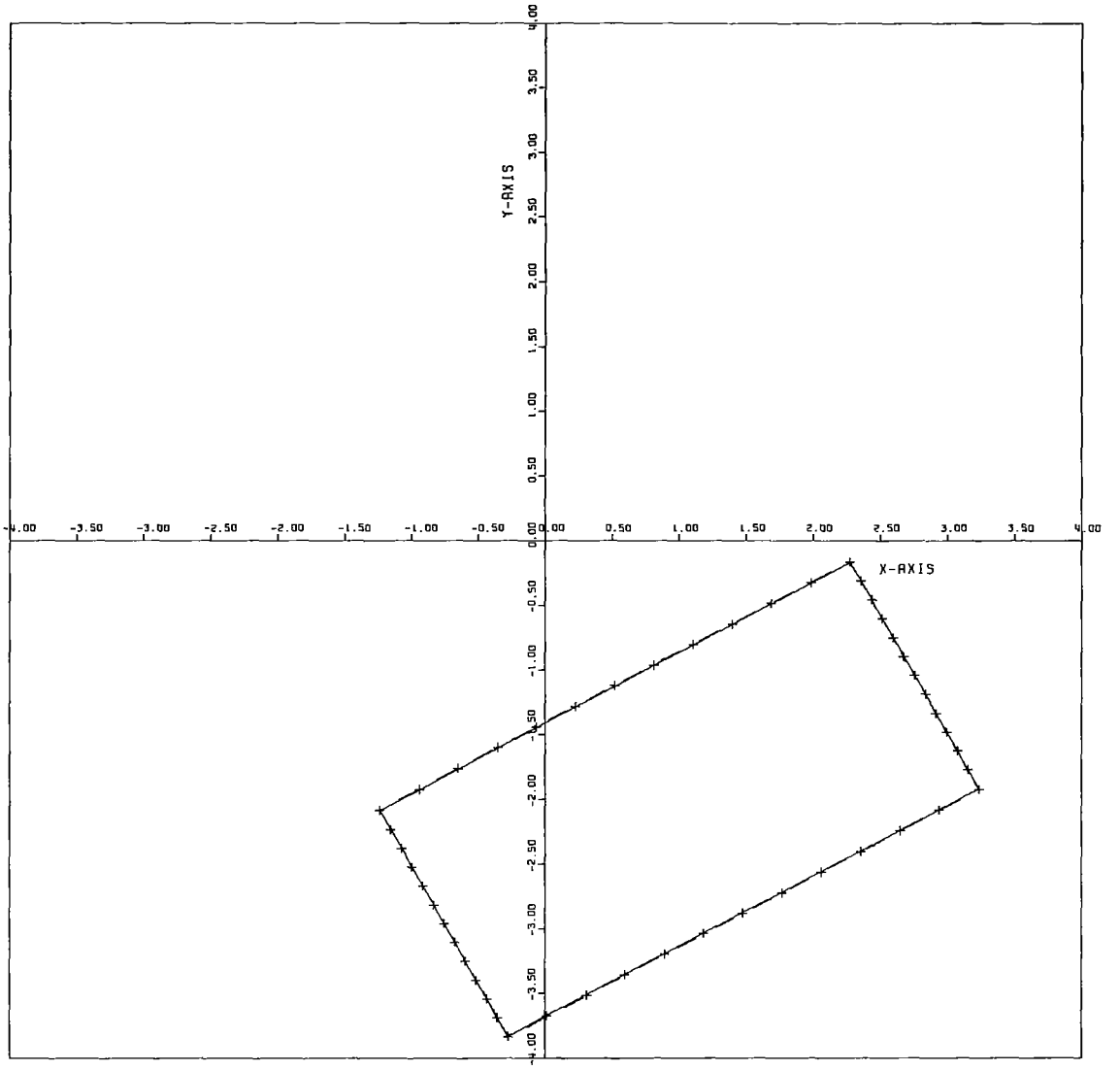
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 1.999
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

Fig. 15--Continued .

FIELD OF VIEW



48 DATA POINTS
STANDARD DEVIATION = 0.0

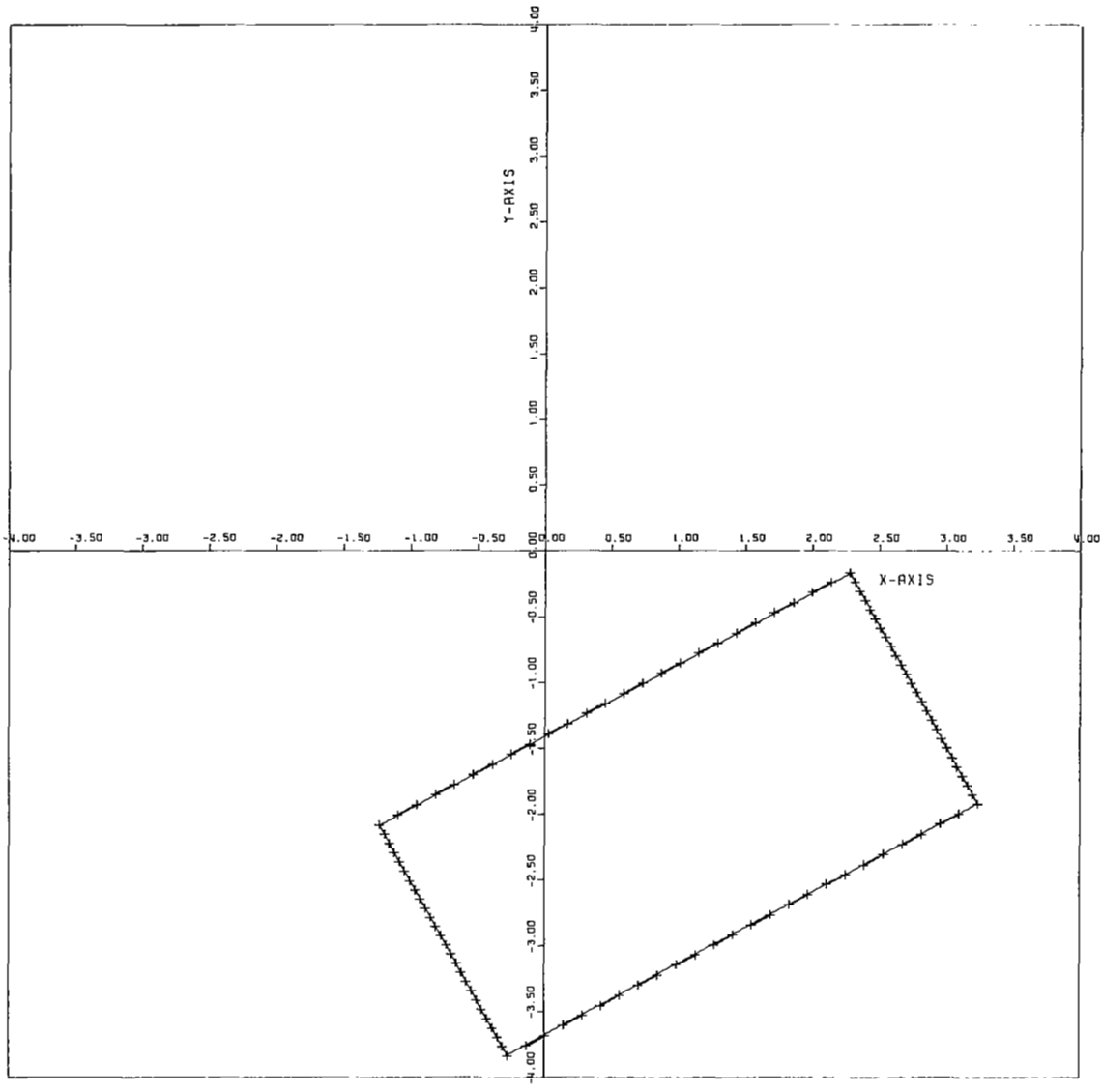
REFERENCE RECTANGLE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 1.999
Y-RADIUS = 0.999
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

Fig. 15--Continued .



100 DATA POINTS
STANDARD DEVIATION = 0.0

REFERENCE RECTANGLE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.001
Y-RADIUS = 1.001
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

Fig. 15--Concluded .

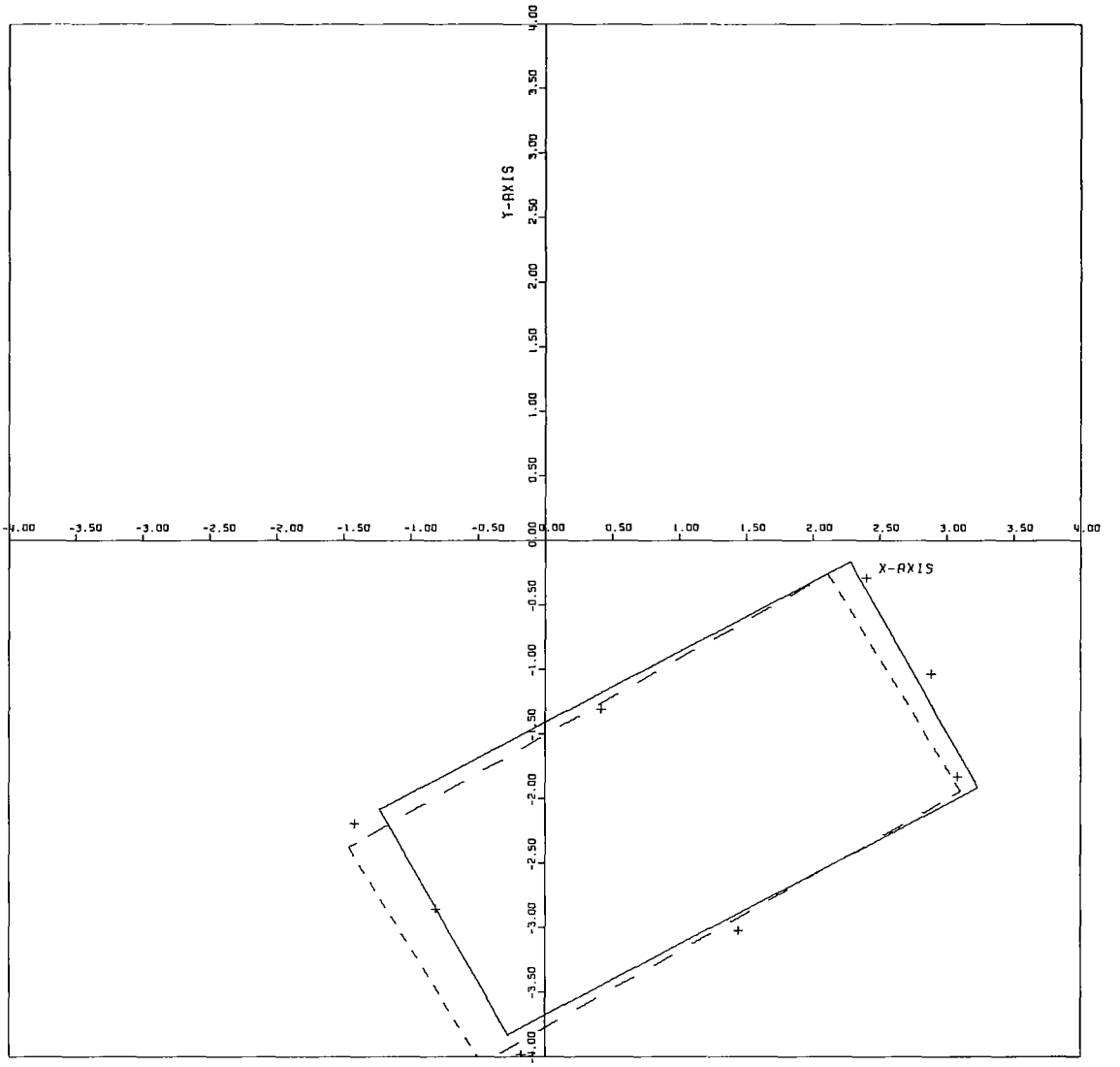
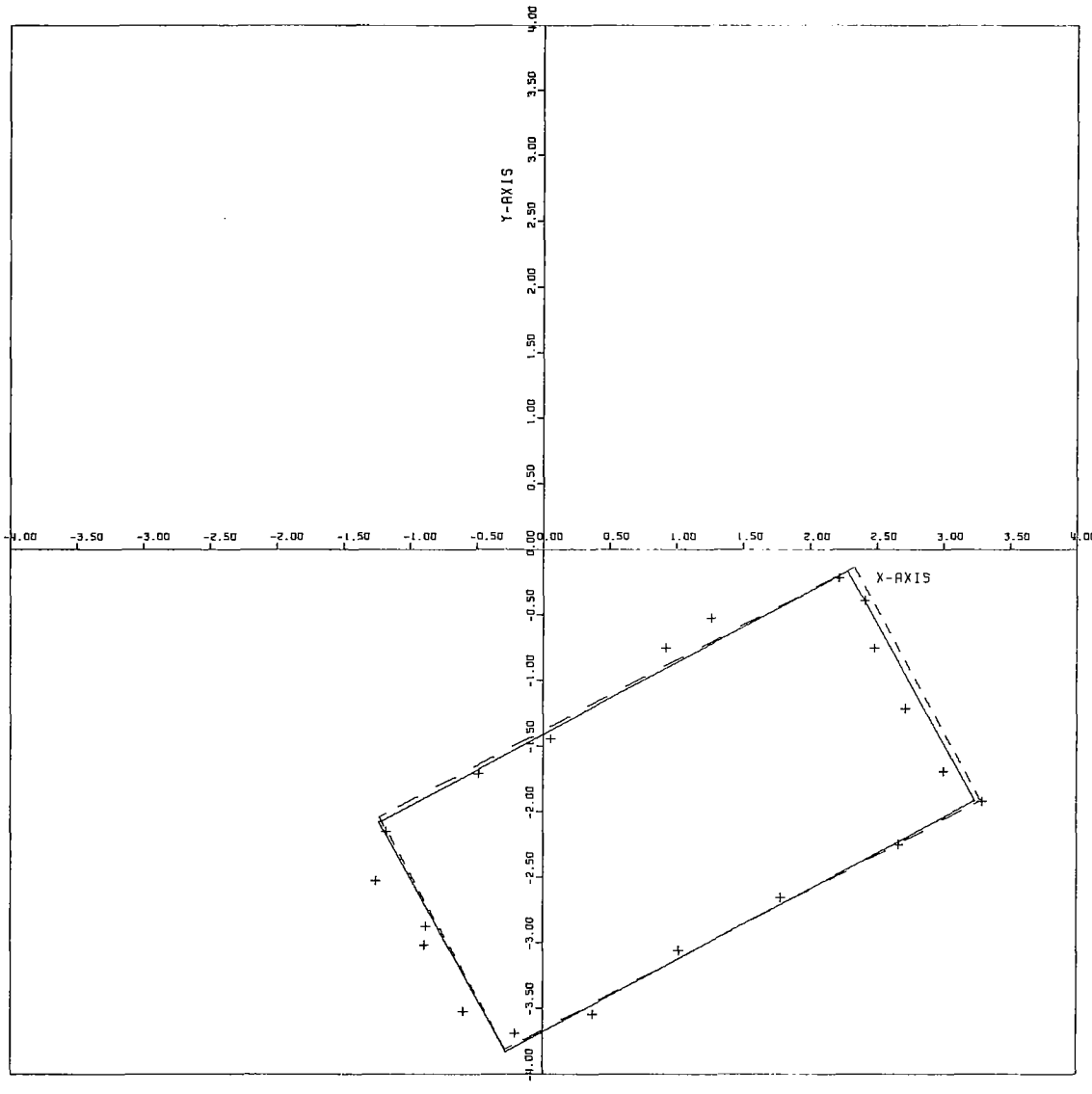


Fig. 16--Rectangles fitted to data points, $\sigma = 0.1$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.1

REFERENCE RECTANGLE

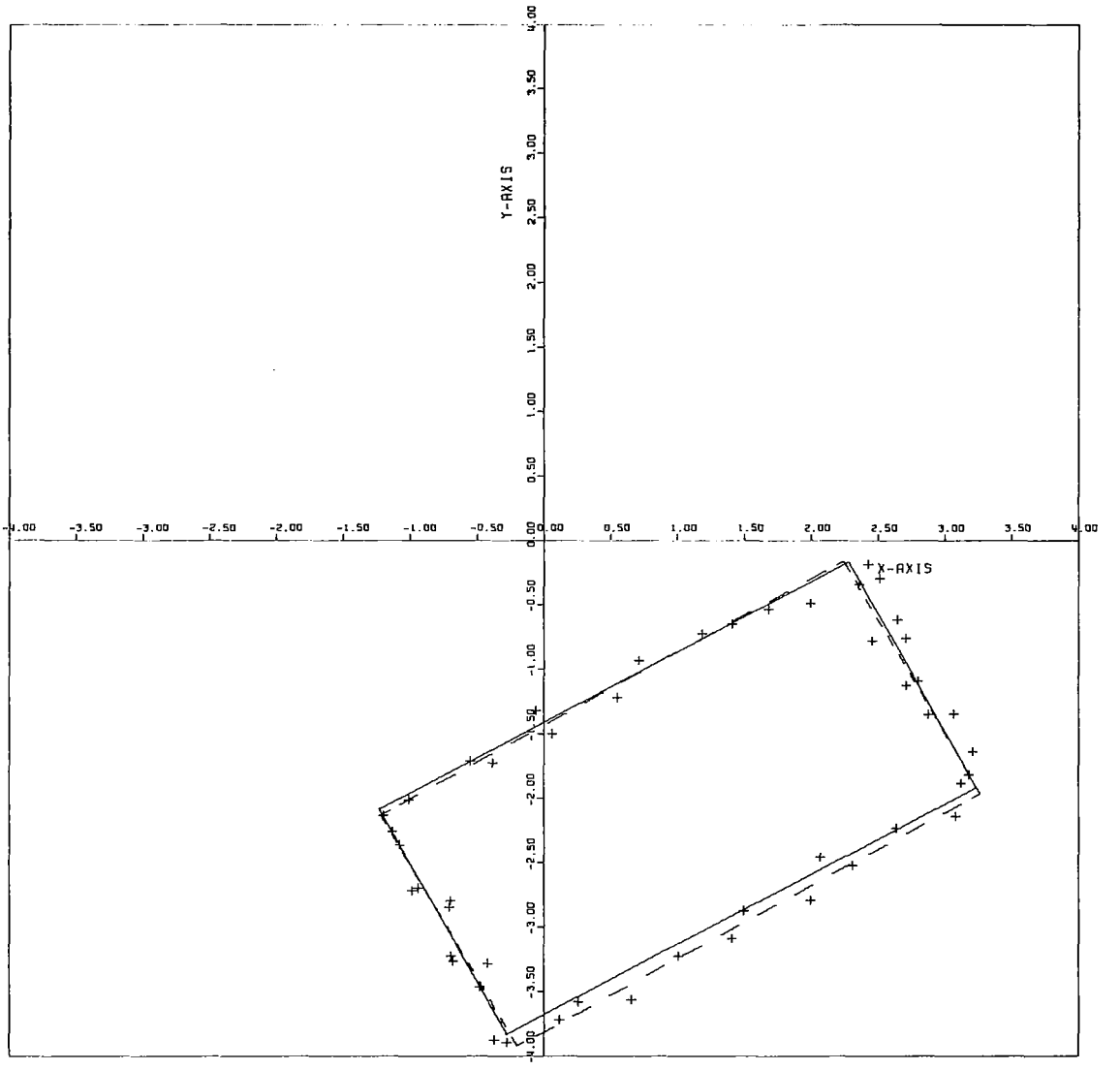
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.020
Y-RADIUS = 1.004
X-TRANSLATION = 1.023
Y-TRANSLATION = -1.977
ROTATION IN DEGREES = 28.137

Fig. 16--Continued .

FIELD OF VIEW



48 DATA POINTS
STANDARD DEVIATION = 0.1

REFERENCE RECTANGLE

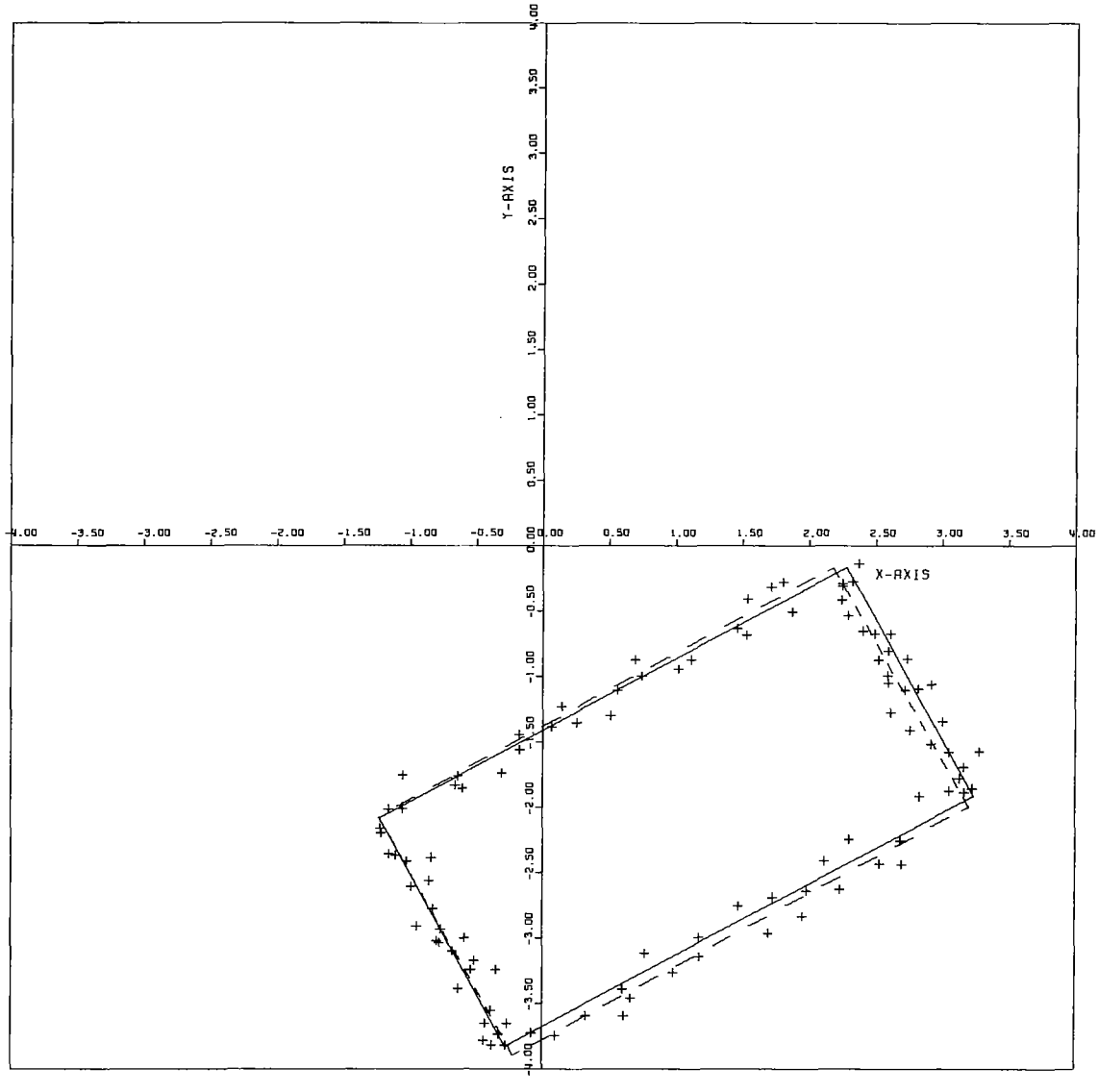
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 1.992
Y-RADIUS = 1.031
X-TRANSLATION = 1.021
Y-TRANSLATION = -2.044
ROTATION IN DEGREES = 29.634

Fig. 16--Continued .

FIELD OF VIEW



100 DATA POINTS
STANDARD DEVIATION = 0.1

REFERENCE RECTANGLE

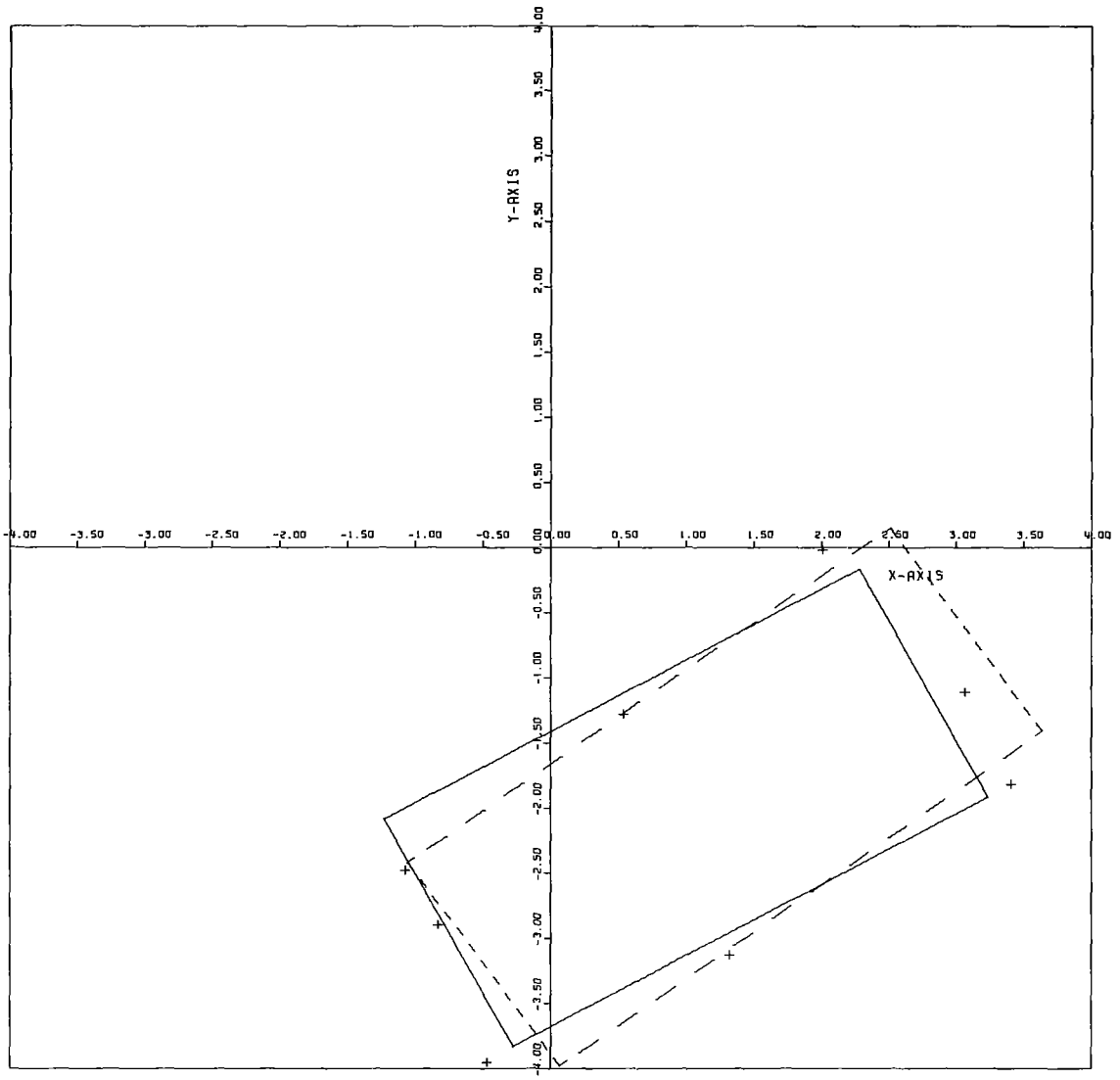
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 1.957
Y-RADIUS = 1.042
X-TRANSLATION = 0.979
Y-TRANSLATION = -2.033
ROTATION IN DEGREES = 29.195

Fig. 16--Concluded .

FIELD OF VIEW



8 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE RECTANGLE

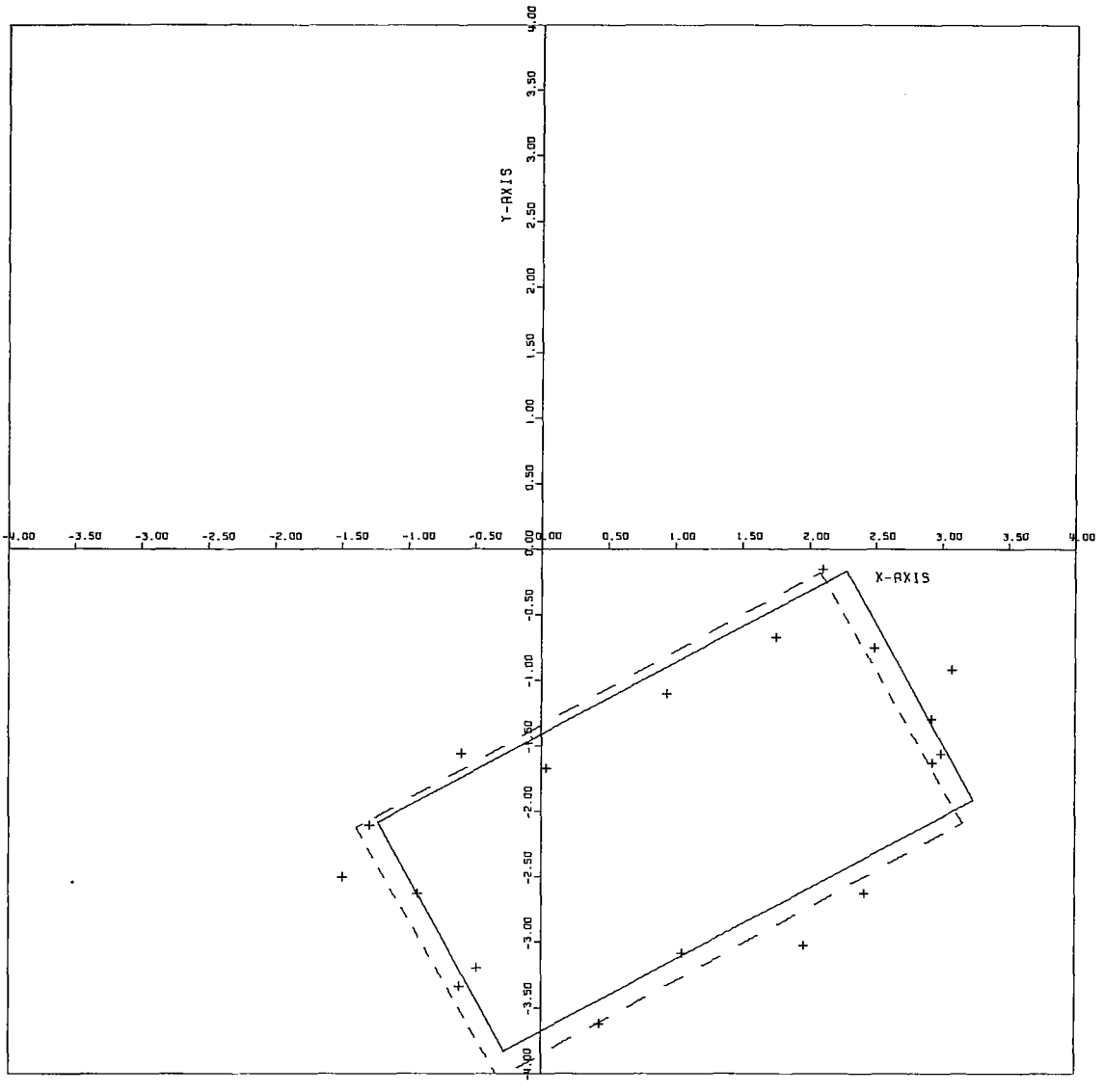
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.200
Y-RADIUS = 0.960
X-TRANSLATION = 1.289
Y-TRANSLATION = -1.917
ROTATION IN DEGREES = 35.822

Fig. 17--Rectangles fitted to data points, $\sigma = 0.2$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE RECTANGLE

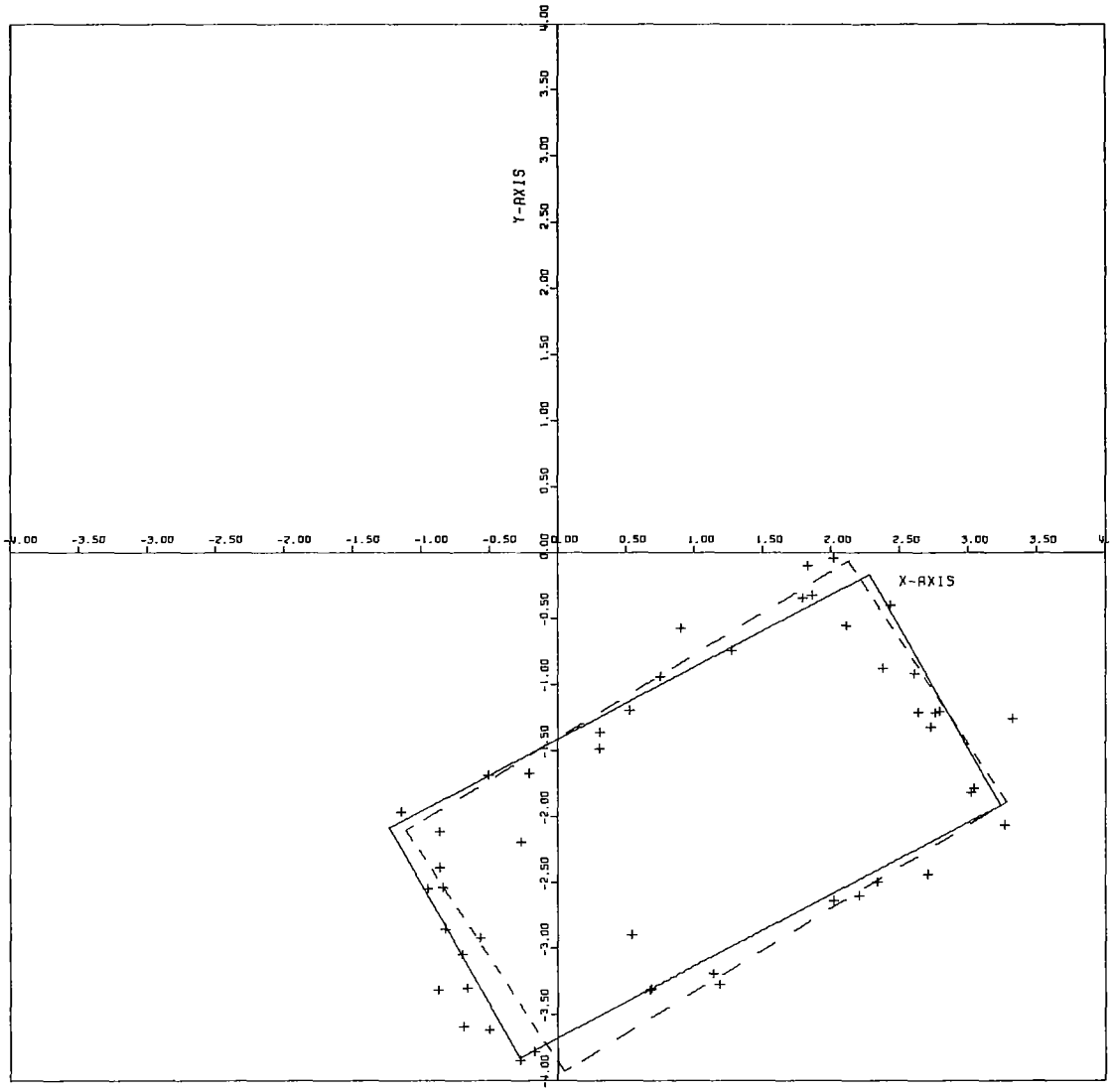
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 1.993
Y-RADIUS = 1.092
X-TRANSLATION = 0.878
Y-TRANSLATION = -2.107
ROTATION IN DEGREES = 29.308

Fig. 17--Continued .

FIELD OF VIEW



48 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE RECTANGLE

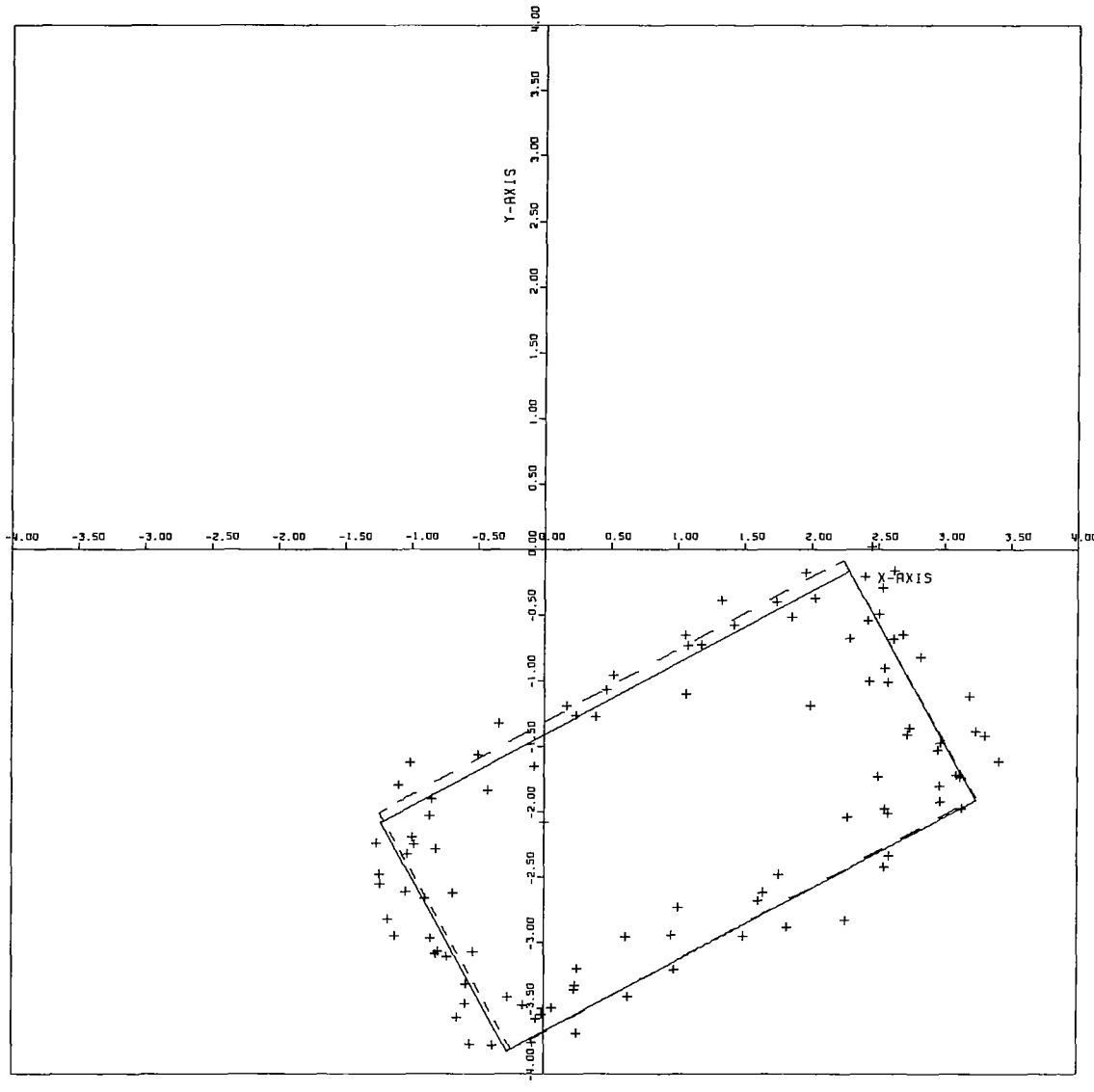
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 1.914
Y-RADIUS = 1.075
X-TRANSLATION = 1.084
Y-TRANSLATION = -1.998
ROTATION IN DEGREES = 32.288

Fig. 17--Continued .

FIELD OF VIEW



100 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE RECTANGLE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 1.990
Y-RADIUS = 1.031
X-TRANSLATION = 0.993
Y-TRANSLATION = -1.952
ROTATION IN DEGREES = 28.954

Fig. 17--Concluded .

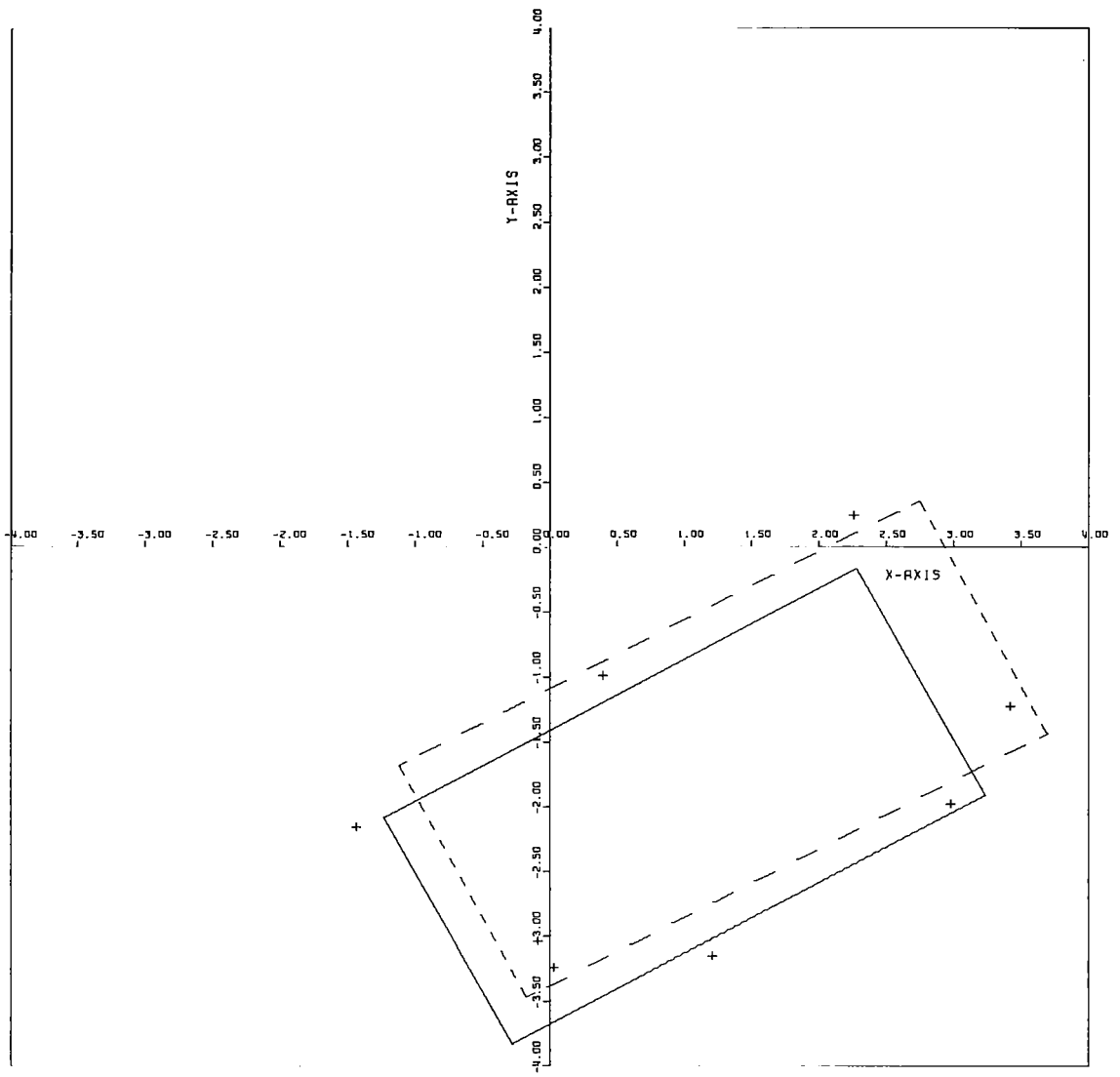
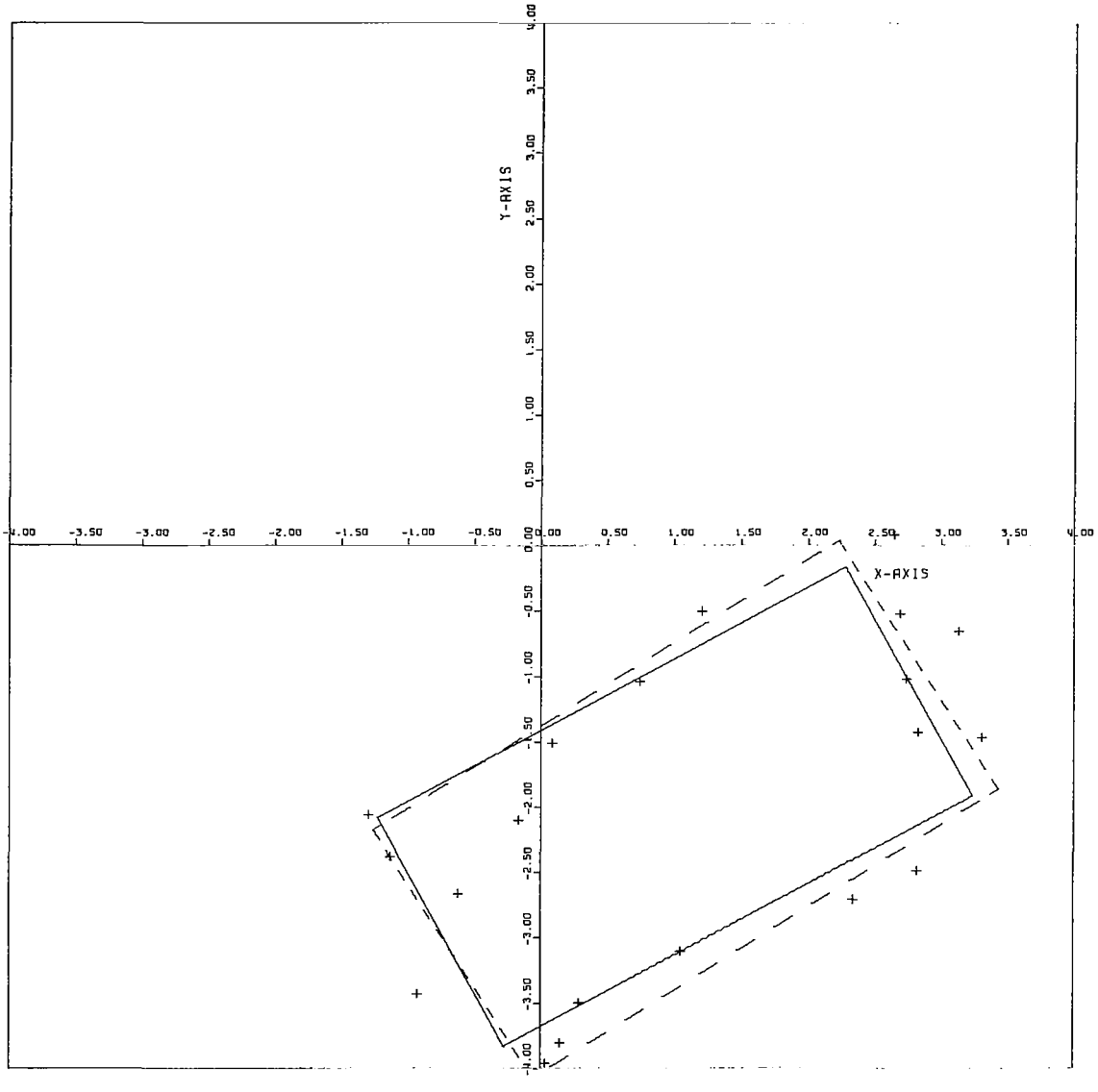


Fig. 18--Rectangles fitted to data points, $\sigma = 0.3$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.3

REFERENCE RECTANGLE

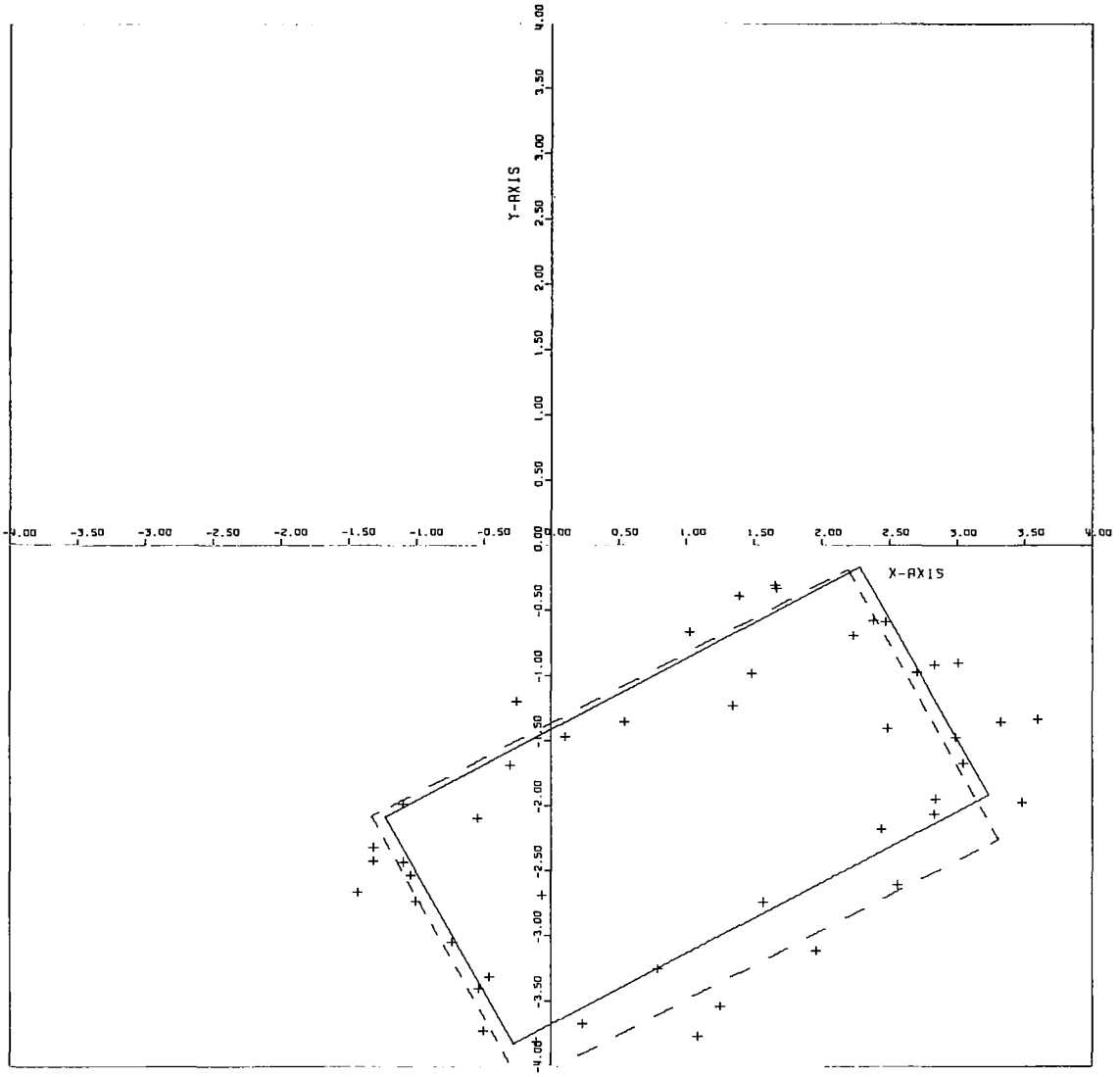
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.069
Y-RADIUS = 1.118
X-TRANSLATION = 1.082
Y-TRANSLATION = -2.020
ROTATION IN DEGREES = 32.435

Fig. 18--Continued .

FIELD OF VIEW



48 DATA POINTS
STANDARD DEVIATION = 0.3

REFERENCE RECTANGLE

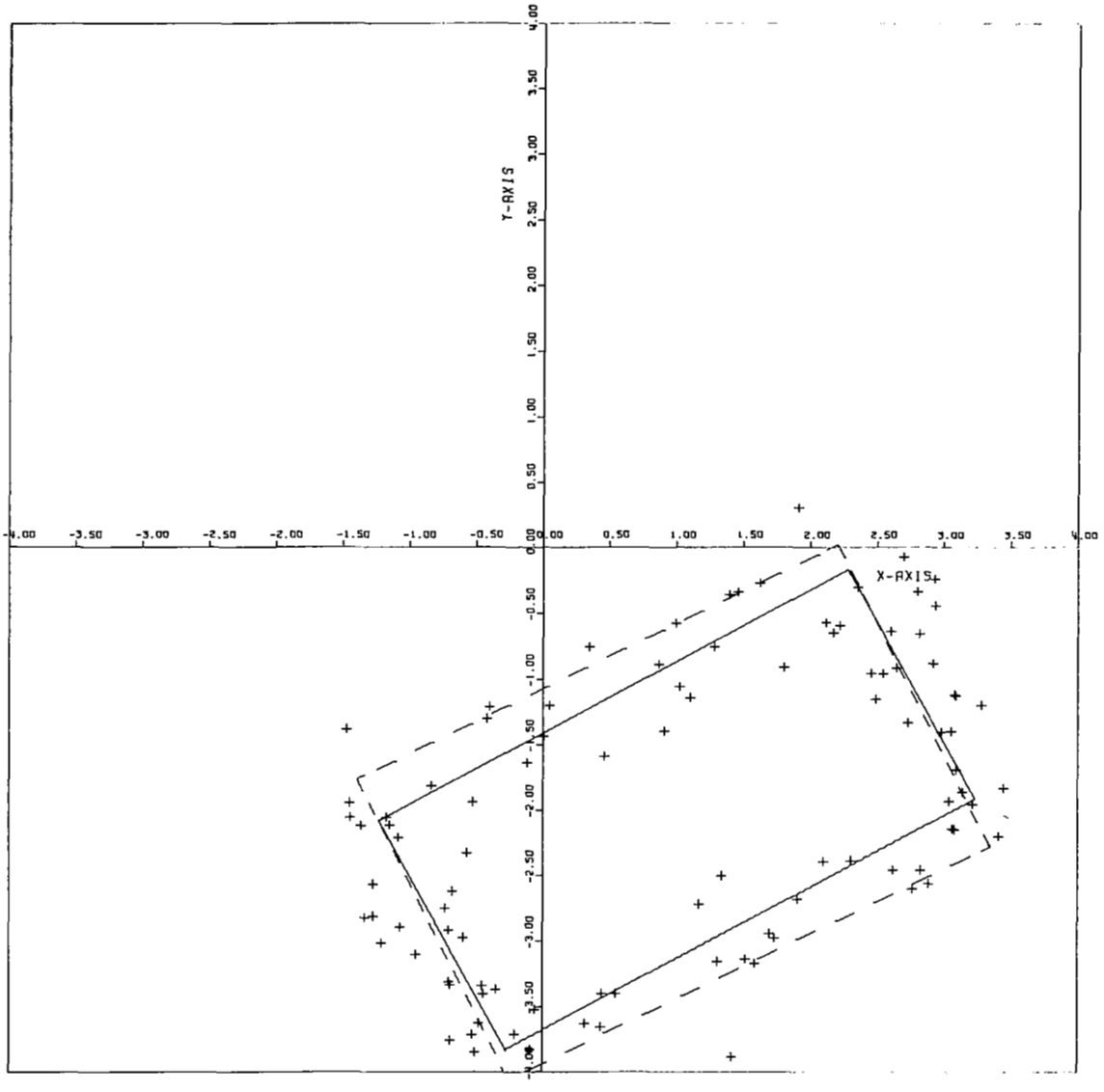
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.002
Y-RADIUS = 1.168
X-TRANSLATION = 0.985
Y-TRANSLATION = -2.167
ROTATION IN DEGREES = 28.298

Fig. 18--Continued .

FIELD OF VIEW



100 DATA POINTS
STANDARD DEVIATION = 0.3

REFERENCE RECTANGLE

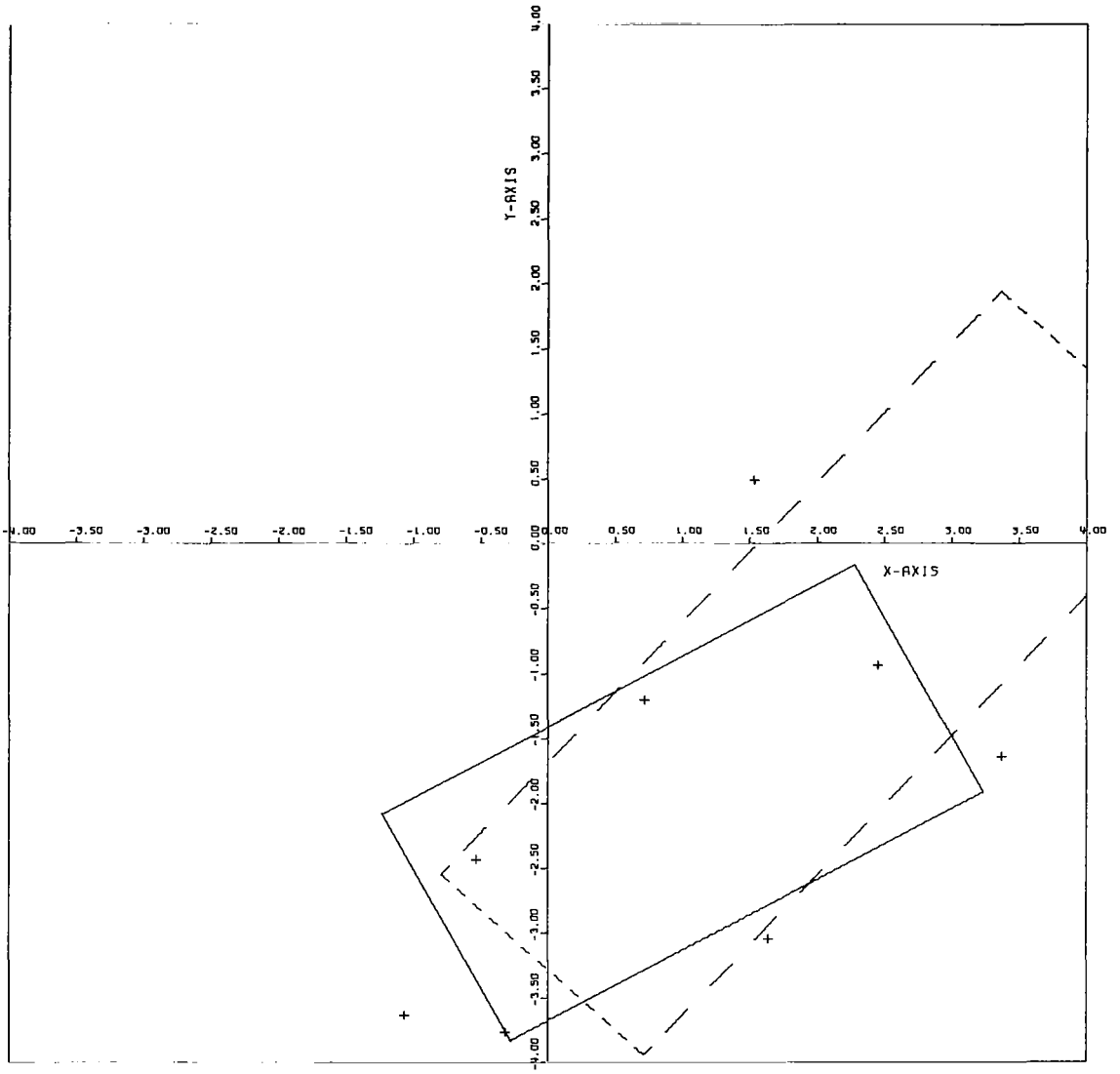
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.010
Y-RADIUS = 1.277
X-TRANSLATION = 0.975
Y-TRANSLATION = -2.022
ROTATION IN DEGREES = 26.393

Fig. 18--Concluded .

FIELD OF VIEW



8 DATA POINTS
STANDARD DEVIATION = 0.4

REFERENCE RECTANGLE

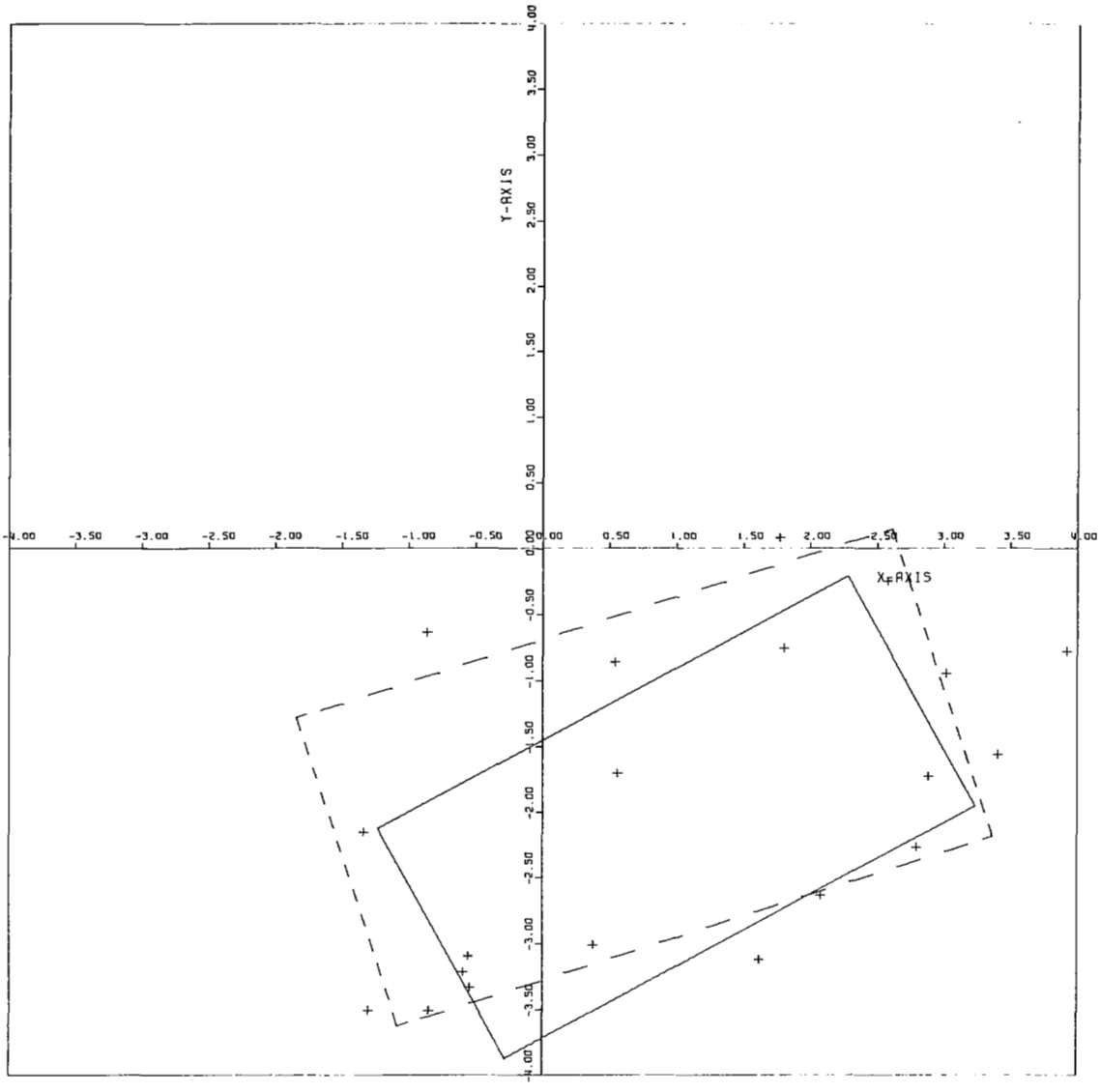
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 3.062
Y-RADIUS = 1.026
X-TRANSLATION = 2.039
Y-TRANSLATION = -1.004
ROTATION IN DEGREES = 47.132

Fig. 19--Rectangles fitted to data points, $\sigma = 0.4$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.4

REFERENCE RECTANGLE

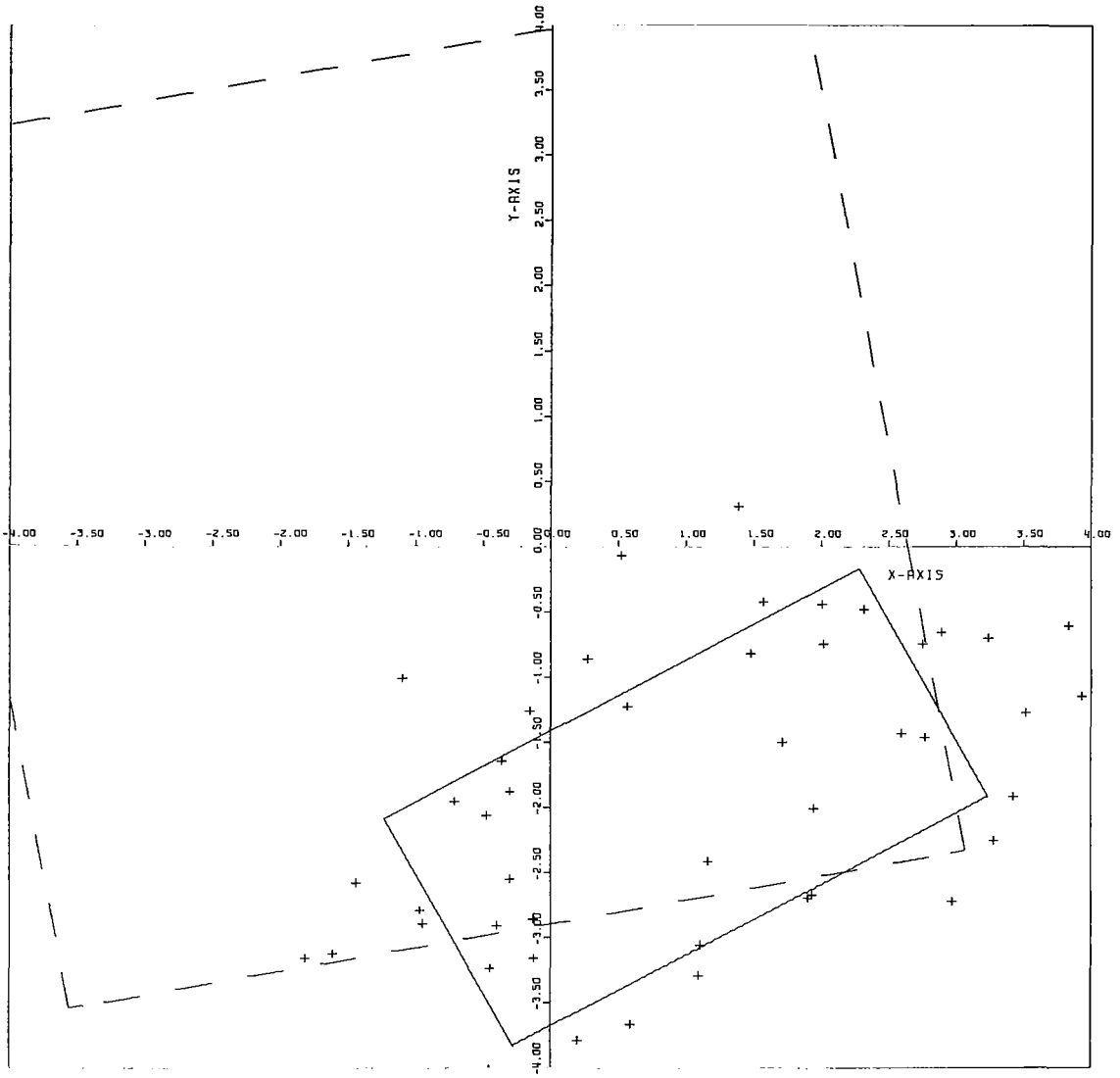
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.338
Y-RADIUS = 1.226
X-TRANSLATION = 0.763
Y-TRANSLATION = -1.697
ROTATION IN DEGREES = 17.918

Fig. 19--Continued .

FIELD OF VIEW



48 DATA POINTS
STANDARD DEVIATION = 0.4

REFERENCE RECTANGLE

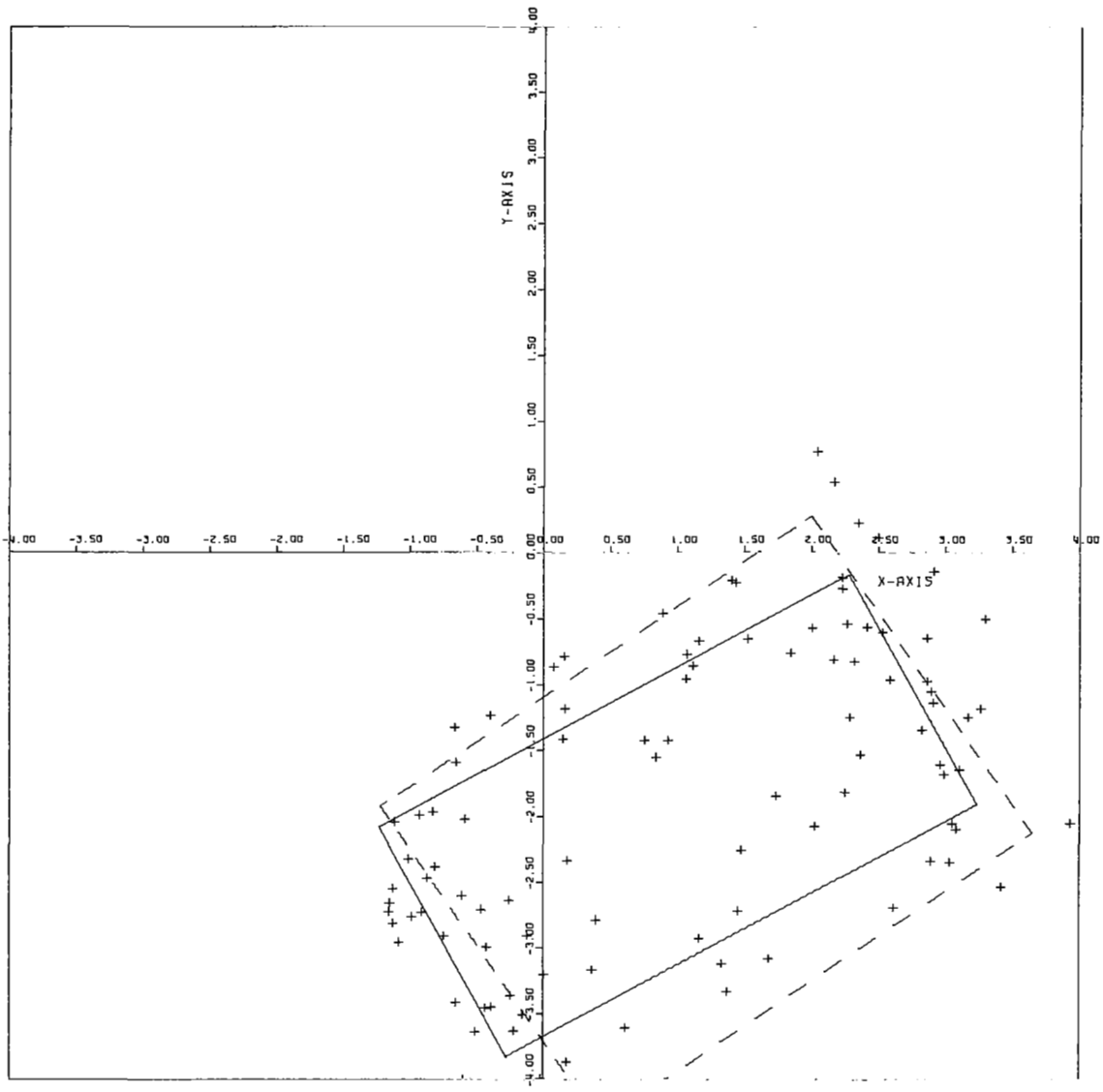
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 3.368
Y-RADIUS = 3.368
X-TRANSLATION = -0.857
Y-TRANSLATION = 0.381
ROTATION IN DEGREES = 10.442

Fig. 19--Continued .

FIELD OF VIEW



100 DATA POINTS
STANDARD DEVIATION = 0.4

REFERENCE RECTANGLE

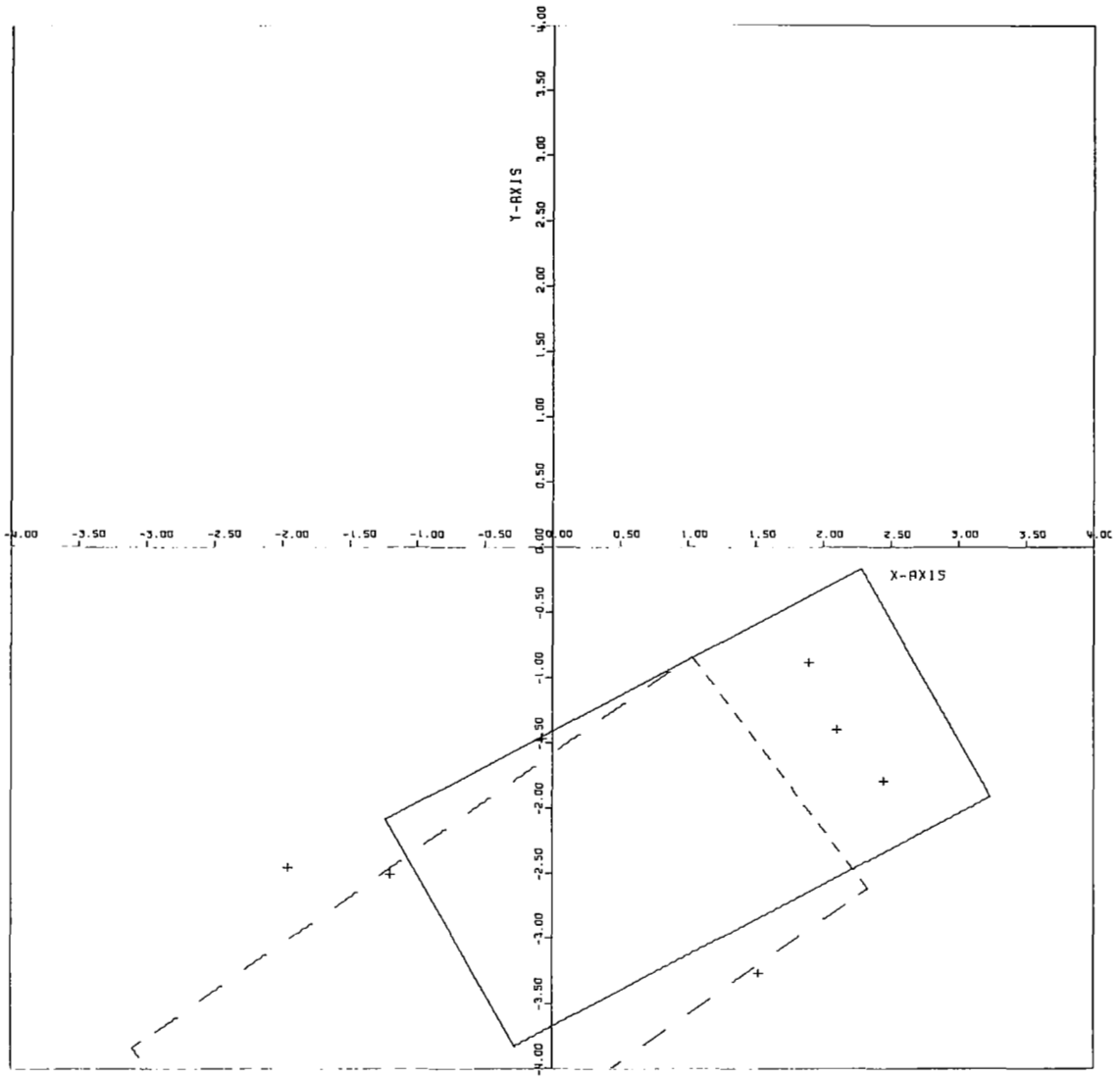
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 1.954
Y-RADIUS = 1.452
X-TRANSLATION = 1.213
Y-TRANSLATION = -2.033
ROTATION IN DEGREES = 34.459

Fig. 19--Concluded .

FIELD OF VIEW



8 DATA POINTS
STANDARD DEVIATION = 0.5

REFERENCE RECTANGLE

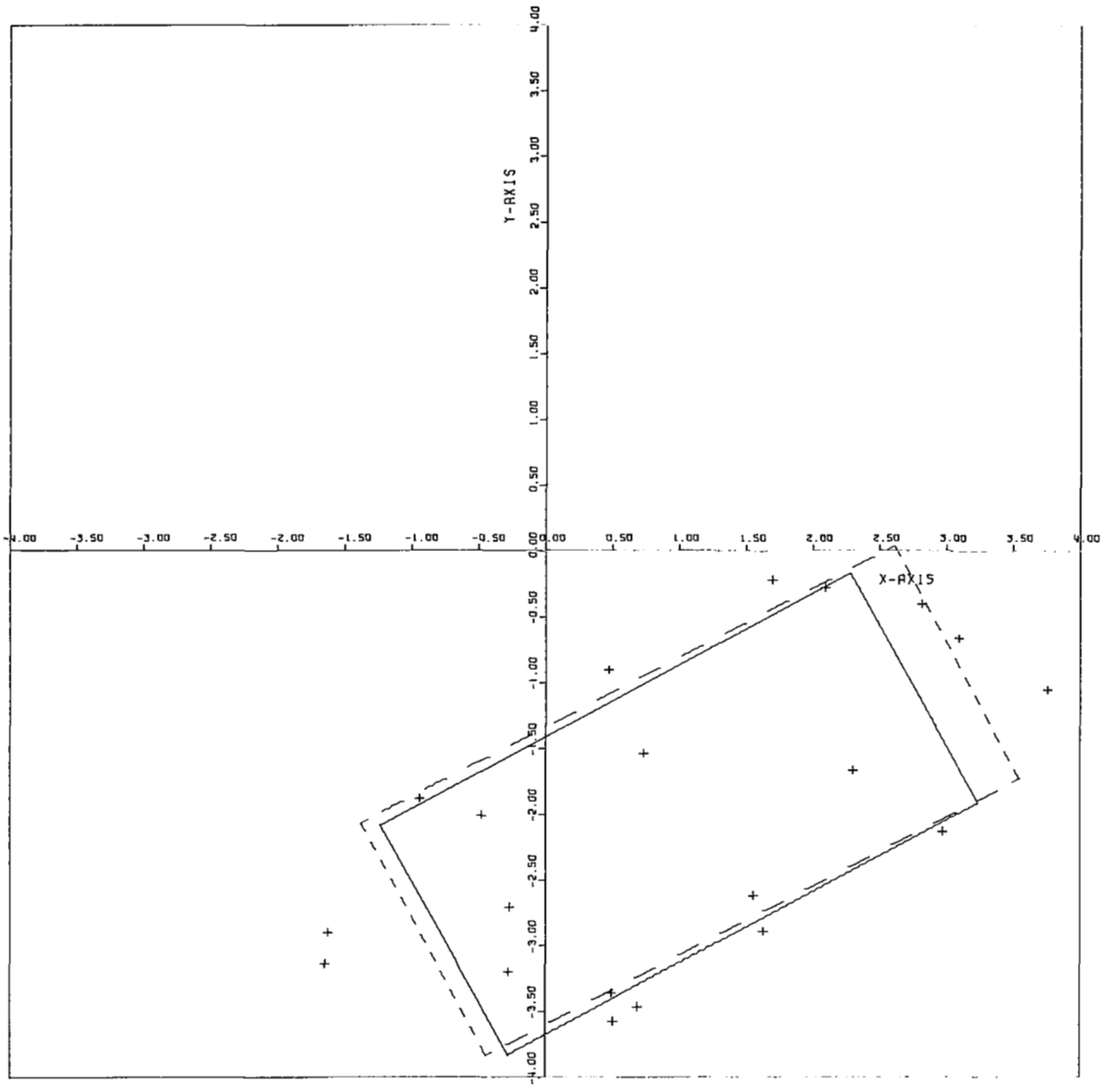
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.559
Y-RADIUS = 1.098
X-TRANSLATION = -0.386
Y-TRANSLATION = -3.234
ROTATION IN DEGREES = 36.110

Fig. 20--Rectangles fitted to data points, $\sigma = 0.5$.

FIELD OF VIEW



20 DATA POINTS
STANDARD DEVIATION = 0.5

REFERENCE RECTANGLE

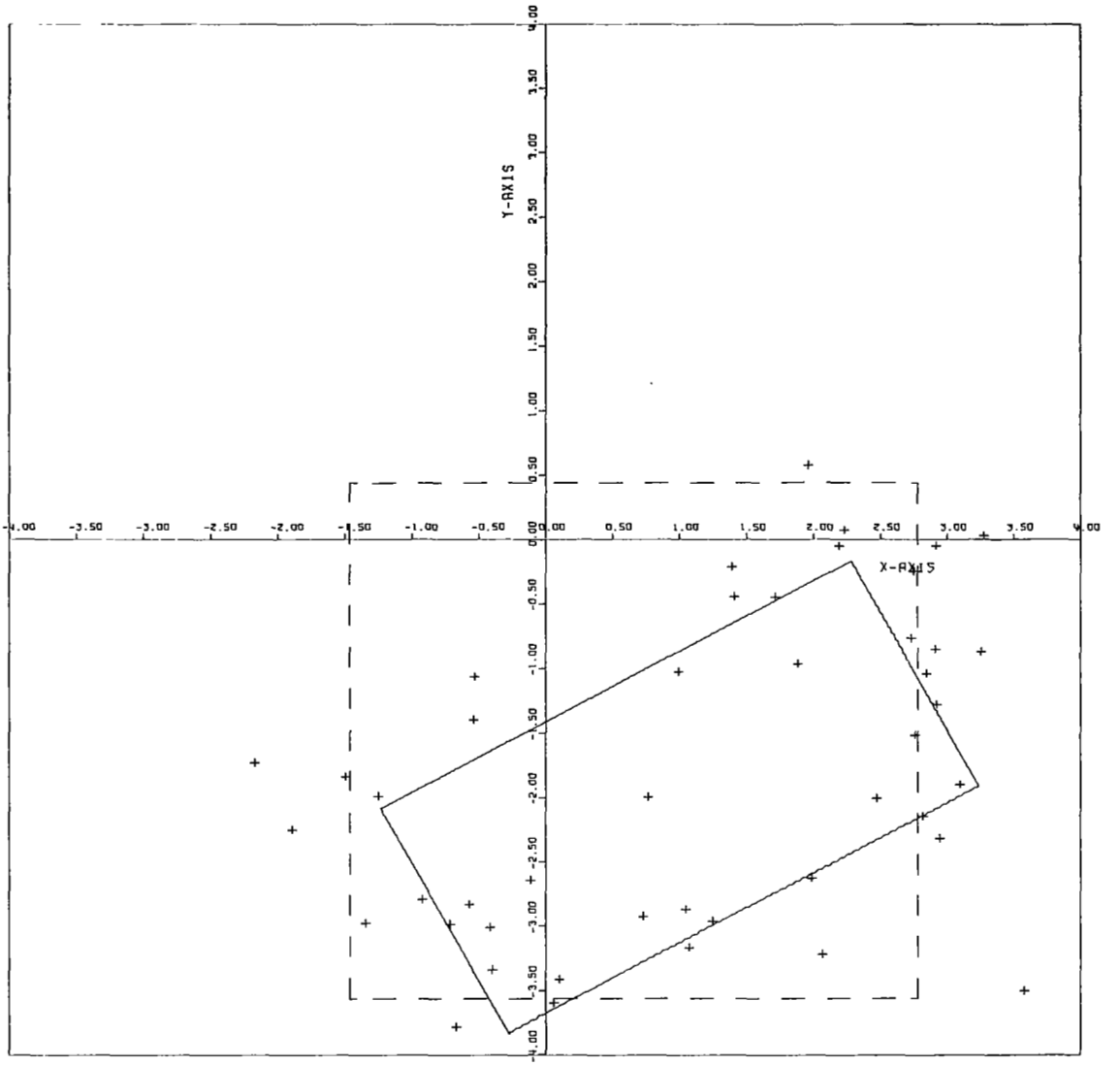
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.258
Y-RADIUS = 0.995
X-TRANSLATION = 1.082
Y-TRANSLATION = -1.899
ROTATION IN DEGREES = 27.832

Fig. 20--Continued .

FIELD OF VIEW



48 DATA POINTS
STANDARD DEVIATION = 0.5

REFERENCE RECTANGLE

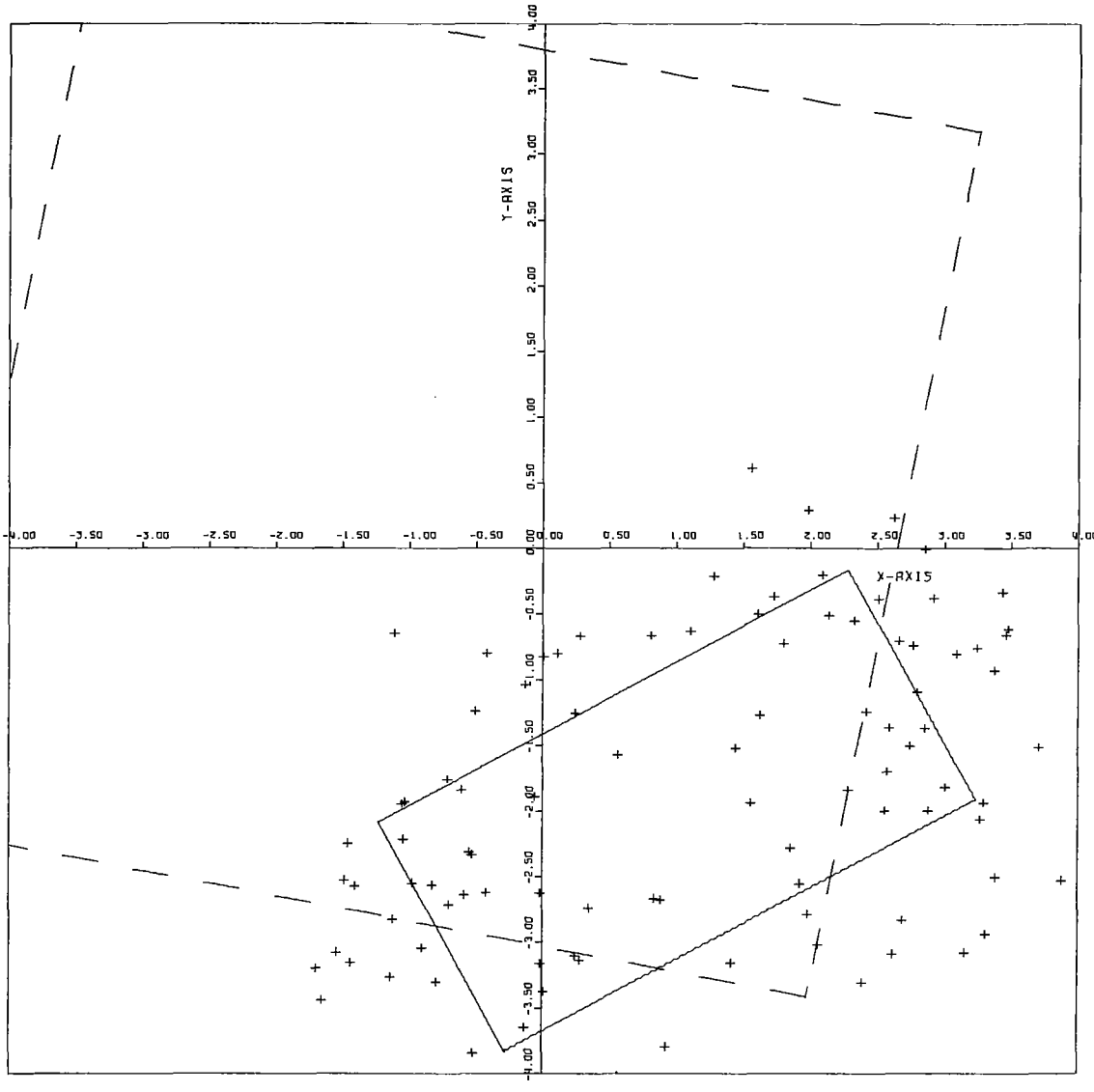
X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 2.120
Y-RADIUS = 2.000
X-TRANSLATION = 0.659
Y-TRANSLATION = -1.560
ROTATION IN DEGREES = 0.100

Fig. 20--Continued .

FIELD OF VIEW



100 DATA POINTS
STANDARD DEVIATION = 0.5

REFERENCE RECTANGLE

X-RADIUS = 2.000
Y-RADIUS = 1.000
X-TRANSLATION = 1.000
Y-TRANSLATION = -2.000
ROTATION IN DEGREES = 28.648

LEAST-SQUARES RECTANGLE

X-RADIUS = 3.358
Y-RADIUS = 3.380
X-TRANSLATION = -0.701
Y-TRANSLATION = 0.516
ROTATION IN DEGREES = 79.057

Fig. 20--Concluded .

CHAPTER VI

SUMMARY AND CONCLUSIONS

This investigation has been aimed at a solution of the problem of real time landmark identification from spacecraft optical fields. The approach which has been taken relies upon the reduction of two dimensional optical images to a discrete set of data points associated with the boundary of an object to be identified. Granted such a set of points, the work reported here is directed toward the fitting of computationally generated images to the real image points by means of an algorithm based upon nonlinear regression analysis.

In order to obtain some concrete results, the present investigation has been limited to a consideration of arbitrary elliptical and arbitrary rectangular objects. While such objects may be relatively rare among all possible landmarks of interest, the approach taken is one of approximation of irregular objects by elliptical or rectangular templates. That is, the methods developed are tolerant of large amounts of noise whether this noise is introduced by measurement and sensing processes or by the deviation of real objects from elliptical or rectangular shapes. Thus, if a known object is within the field of view, precise information about its size, location, and orientation can be obtained, for example, from the parameters of the least squares ellipse fitted to its boundary points.

The computational procedures described in this report are totally insensitive to image rotation, translation, and scale change. So far as is known to the authors, no alternative image processing technique exists with a capability of producing extremely accurate object parameter estimates under such conditions. The algorithms presented are thus felt to provide the first feasible method for the generation of very precise navigational information from the optical images of known landmarks.

While all of the results contained in this report relate to two dimensional objects, it appears that the basic approach is applicable to three dimensional image analysis as well. Specifically, it seems feasible that a computational procedure could be developed which would be capable of producing a replica of the image of a given three dimensional object produced by a particular optical system with a specified spatial relationship to the object. From such a synthetic image, it ought to be straight-

forward to extract boundary points which could then be compared to the boundary points of the real scene. Iterative adjustment of the spatial parameters, used to generate synthetic images could then be accomplished by the nonlinear regression program included in this report so as to optimize the fit of the synthetic image to the real image. Such an extension of the present work would permit the use of optically derived guidance information in such difficult tasks as automatic orbital rendezvous and docking of spacecraft.

In summary, the ability of a digital computer to extract accurate guidance and control signals from optical fields has been established by this study. Additional work along the lines indicated by this research should eventually produce a very valuable means for a spacecraft or robot vehicle to obtain quantitative information regarding its position and angular orientation relative to objects within its field of view.

APPENDIX I

GENERATION OF DATA POINTS

The data points which are presented to the parameter estimation algorithm are not physically measured points since no equipment was available for this purpose. The entire data acquisition process is instead simulated by a digital computer. The following sections briefly explain how the data points which lie on the boundary of an ellipse or a rectangle are generated, as well as how noisy data points may be generated. The subroutine which generated the data points is denoted by DATA.

Ellipse

The generation of the data points lying on the boundary of an ellipse shall be considered first. An ellipse, as shown in Figure A-1, may be expressed analytically by Eqn. A-1.

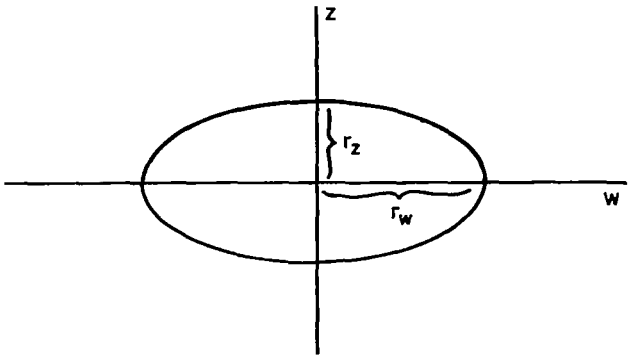


Fig. A-1--Ellipse in w, z-reference frame.

$$e_1 w^2 + e_2 z^2 = 1 \qquad \text{(A-1)}$$

where

$$e_1 = \frac{1}{r_w^2} \quad \text{and} \quad e_2 = \frac{1}{r_z^2} \qquad \text{(A-2)}$$

Thus, if one is given the two parameters e_1 and e_2 then the corresponding ellipse in the w-z plane is completely specified. It is desired to represent this ellipse's boundary by some finite number of points. For a given number of data points, say N, there are infinitely many different ways in which these points may be positioned on the ellipse's boundary. However, it seems quite unrealistic to have the data points very dense on one portion of the boundary and very sparse, or nonexistent, on the remaining portion of the boundary. Perhaps the most realistic situation is for the data points to be uniformly distributed on the boundary of the ellipse. This would require the distance between any two adjacent data points to be L/N, where L is the length of the boundary of the ellipse. This particular distribution of the data points on the boundary of the ellipse was not used, however, because of the complex computer programming which would be involved and because in practice the physical measuring equipment probably would not select the data points in precisely this manner anyway.

The method which was employed to select data points on the boundary of the ellipse consists of dividing the w-axis diameter of the ellipse into N/2 equal length segments when N data points are desired. This, of course, requires N to be an even number. This procedure is shown in Figure A-2 for N = 10.

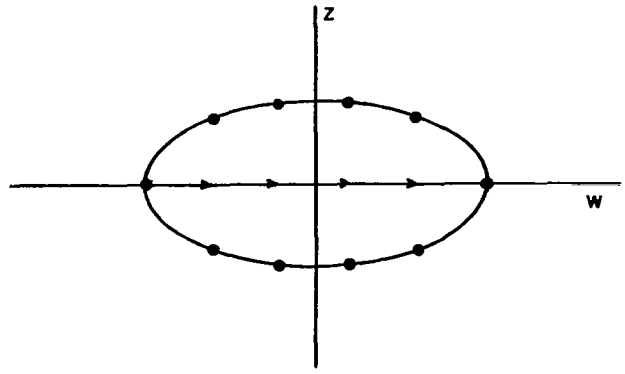


Fig. A-2--Data points corresponding to ellipse in w, z-reference frame.

The N data points then comprise those points on the boundary of the ellipse whose w-coordinates are the same as the w-coordinates of the end points of the segments of the w-axis diameter.

After the w, z-coordinates of the data points which lie on the boundary of the ellipse have been determined, it is necessary to find the coordinates of these same data points with respect to the reference x, y-coordinate system. These two coordinate systems are shown in Figure A-3.

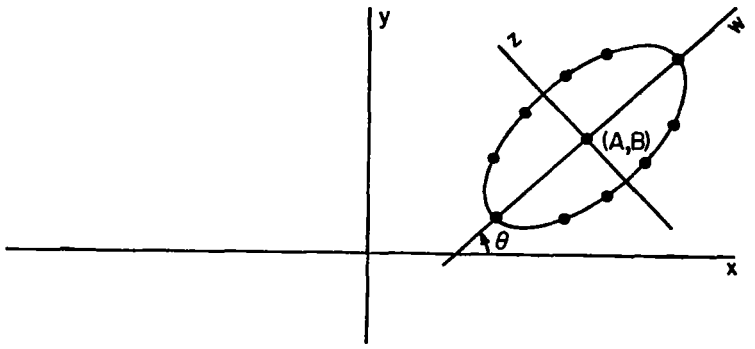


Fig. A-3--Data points corresponding to ellipse in x, y-reference frame.

Once the x and y-translation (denoted by A and B, respectively) of the center of the ellipse and the rotation (denoted by θ) of the w-axis of the ellipse are specified, then the x, y-coordinates of the data point having w, z-coordinates (w_i, z_i) are

$$x_i = w_i \cos \theta - z_i \sin \theta + A \qquad (A-3)$$

$$y_i = w_i \sin \theta + z_i \cos \theta + B \qquad (A-4)$$

The x, y-coordinates of the data points are then taken as the coordinates of the data points which represent the ellipse whose parameters are now to be estimated.

Rectangle

The data points which lie on the boundary of a given rectangle are generated in a slightly different manner than those of an ellipse. Figure A-4 shows a rectangle whose center is at the origin of the w, z-coordinate system.

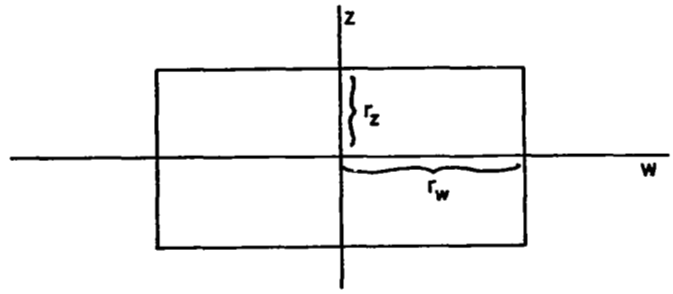


Fig. A-4--Rectangle in w, z-reference frame.

The rectangle's dimension in the w-direction is $2r_w$, while its dimension in the z-direction is $2r_z$. The dimensions r_w and r_z may be thought of as "radii" of the rectangle.

The data points are selected such that one quarter of the total number of data points lie on each side of the rectangle. This requires N to be devisible by four. The data points are further restricted to be equally spaced along each side. Therefore, the spacing between data points which lie on the vertical boundaries is $8r_z/N$ while the spacing between data points which lie on the horizontal boundaries is $8r_w/N$. Once the w, z-coordinates of all the data points lying on the boundary of the rectangle are found, their corresponding x, y-coordinates may be determined from Eqns. A-3, 4.

Noise

The preceeding discussion has briefly explained how the data points corresponding to either an ellipse or a rectangle are generated, being given the parameters e_1, e_2, A, B, θ or r_w, r_z, A, B, θ. These data points fall exactly on the boundary of the appropriate pattern. Since there is no error in the coordinates of these data points, they may be considered as noiseless data points.

In a realistic system, however, one would expect that the measurement points would not exactly overlay the boundary of the pattern from which they came. This error may be due to several different reasons. For instance, if the field of view has been slightly clouded over, or defocused, then the boundary of the pattern is no longer precise and the exact coordinates of points lying on the boundary can only be estimated. Even if the field of view is clear there is still the possibility that the electronic equipment associated with the optical system can commit errors, be they internal or transmission errors. Furthermore,

there is always the quantization error associated with analog to digital conversion. All of these errors may be considered as forms of noise.

Therefore, in order for the artificially generated data points to realistically correspond to physically measured data points it is necessary to degrade the artificially generated data points by corrupting them with some type of noise. A detailed analysis of the physical system would be required in order to know the exact nature of the actual noise; i.e., its distribution and whether it is additive, multiplicative, or whatever. In this study no particular physical system was considered; therefore, the noise samples were assumed to be additive, statistically independent, gaussian noise samples. How well this artificial noise resembles the actual noise in a physical system was not considered.

The gaussian noise was generated by the subroutine GAUSS which is in the library of the IBM 360/75 at The Ohio State University Computer Center. The subroutine permits one to specify both the mean and the standard deviation of the gaussian noise samples which it is to generate. The subroutine makes use of another library subroutine called RANDU which generates uniformly distributed random numbers in the range 0-1. The subroutine GAUSS approximates a gaussian random variable by adding together twelve uniform random variables, making use of the Central Limit Theorem.

Since it is assumed that the noise is additive gaussian, the numbers which are generated by GAUSS are simply added independently to each coordinate of the data points. Thus, if the coordinates of a noiseless data point are given by (X_i, Y_i), then the coordinates of the corresponding noisy data point are (x_i, y_i), where

$$x_i = X_i + n_j \qquad (A\text{-}5)$$

$$y_i = Y_i + n_{j+1} \qquad (A\text{-}6)$$

where n_j and n_{j+1} are two consecutive noise samples.

APPENDIX II

DISCUSSION OF CRITERION FUNCTIONS

It is important to recognize the fact that the criterion function used in the case where the error function is a linear function of a set of parameters (one step minimization method) is not identical (even to within a scale factor) to the criterion function which results when the error function is a nonlinear function of a different set of parameters (iterative minimization scheme).

To be more specific, suppose that the criterion function resulting from the linear error function (Eqn. 3.53) is denoted by ϕ_L and written as

$$\phi_L(\vec{x}, \vec{y}; \vec{p}) = \bar{e}^T \bar{e}$$

$$= \sum_{i=1}^{N} (p_1 x_i^2 + p_2 x_i y_i + p_3 y_i^2 + p_4 x_i + p_5 y_i + 1)^2 \qquad (A-7)$$

The minimization of ϕ_L is taken with respect to the parameter vector $\vec{p} = (p_1, p_2, p_3, p_4, p_5)^T$. If $\vec{p}^*$ is that value of $\vec{p}$ which results in ϕ_L attaining its unique minimum value, ϕ_L^*, then

$$\phi_L^*(\vec{x}, \vec{y}; \vec{p}^*) = \min_{\vec{p}} \phi_L(\vec{x}, \vec{y}; \vec{p}) \qquad (A-8)$$

The ellipse parameter vector $\vec{c} = (e_1, e_2, A, B, \theta)^T$ is then found from Eqn. (3.42), that is,

$$\vec{c}^* = \vec{\psi}(\vec{p}^*) \qquad (A-9)$$

The criterion function resulting from the nonlinear error function, denoted by ϕ_N, may be written as

$$\phi_N(\vec{x}, \vec{y}; \vec{\rho}(\vec{c})) = \vec{e}^T \vec{e}$$

$$= \sum_{i=1}^{N} (\rho_1 x_i^2 + \rho_2 x_i y_i + \rho_3 y_i^2 + \rho_4 x_i + \rho_5 y_i + \rho_6 + 1)^2 \qquad (A-10)$$

where the vector $\vec{\rho} = (\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6)^T$ is a nonlinear function of the ellipse parameter vector $\vec{c}$ with which the minimization of ϕ_N is taken. The vector $\vec{\rho}(\vec{c})$ is given by Eqns. (3.10 - 3.15).

Note that ϕ_N may be factored

$$\phi_N = \sum_{i=1}^{N} (\rho_6 - 1)^2 \left[\frac{\rho_1}{\rho_6 - 1} x_i^2 + \frac{\rho_2}{\rho_6 - 1} x_i y_i + \frac{\rho_3}{\rho_6 - 1} y_i^2 + \frac{\rho_4}{\rho_6 - 1} x_i + \frac{\rho_5}{\rho_6 - 1} y_i + 1 \right]^2$$

$$= (\rho_6 - 1)^2 \, \phi_L \qquad \text{(A-11)}$$

Since $\vec{\rho}$ is a function of $\vec{p}$ (Eqn. 3.39), the factor $(\rho_6 - 1)^2$ may be represented by some function, $g(\vec{p})$, to give

$$\phi_N(\vec{x}, \vec{y}; \vec{p}) = g(\vec{p}) \, \phi_L(\vec{x}, \vec{y}; \vec{p}) \qquad \text{(A-12)}$$

If $\vec{p}^{**}$ is that value of $\vec{p}$ for which ϕ_N attains its minimum value, denoted by ϕ_N^*, then

$$\phi_N^*(\vec{x}, \vec{y}; \vec{p}^{**}) = \min_{\vec{p}} \phi_N(\vec{x}, \vec{y}; \vec{p}) \qquad \text{(A-13)}$$

and so

$$\phi_N^*(\vec{x}, \vec{y}; \vec{p}^{**}) \leq g(\vec{p}^*) \, \phi_L^*(\vec{x}, \vec{y}; \vec{p}^*) \qquad \text{(A-14)}$$

One should note that if ϕ_N is divided by the factor $(\rho_6 - 1)^2$ before the minimization with respect to $\vec{c}$ is taken, then the two criterion functions would be identical and both minimization techniques would yield the same value for the minimizing parameter vector (assuming no boundaries are encountered).

A comparison of ϕ_L^* and ϕ_N^* is made in Table A. The right hand side of Eqn. (A-14) is also tabulated, being denoted by ϕ^*. The values of the criterion functions are those that were obtained by using the two minimization schemes on an ellipse which is characterized by

$$\vec{c}_0 = \begin{bmatrix} 0.25 \\ 1.00 \\ 1.00 \\ -2.00 \\ 0.50 \end{bmatrix} \qquad \text{(A-15)}$$

The parameter vector estimates which correspond to these values for the criterion functions are shown in Tables 1 and 2 in Chapter IV.

It is interesting to note that in every case $\phi_N^* \leq \phi^*$ in Table A-I, as Eqn. (A-14) implies.

TABLE A-I: COMPARISON OF CRITERION FUNCTIONS

σ	N	ϕ_L^*	ϕ^*	ϕ_N^*
0.0	10	0.1564×10^{-7}	0.2496×10^{-6}	0.6555×10^{-11}
0.0	20	0.2379×10^{-7}	0.3795×10^{-6}	0.1846×10^{-10}
0.0	50	0.6631×10^{-6}	0.1060×10^{-4}	0.7957×10^{-10}
0.0	100	0.8263×10^{-6}	0.1319×10^{-4}	0.4480×10^{-10}
0.1	10	0.1757×10^{-1}	0.3219×10^{0}	0.2911×10^{0}
0.1	20	0.1747×10^{-1}	0.3099×10^{0}	0.2913×10^{0}
0.1	50	0.8430×10^{-1}	0.1313×10^{1}	0.1191×10^{1}
0.1	100	0.1697×10^{0}	0.2710×10^{1}	0.2449×10^{1}
0.2	10	0.2264×10^{-1}	0.3992×10^{0}	0.3467×10^{0}
0.2	20	0.1611×10^{0}	0.2566×10^{1}	0.1687×10^{1}
0.2	50	0.2627×10^{0}	0.5528×10^{1}	0.3989×10^{1}
0.2	100	0.5216×10^{0}	0.1164×10^{2}	0.8214×10^{1}
0.3	10	0.9620×10^{-1}	0.2193×10^{1}	0.1232×10^{1}
0.3	20	0.3844×10^{0}	0.6326×10^{1}	0.2912×10^{1}
0.3	50	0.7211×10^{0}	0.1151×10^{2}	0.6331×10^{1}
0.3	100	0.1881×10^{1}	0.3245×10^{2}	0.1480×10^{2}
0.4	10	0.2027×10^{0}	0.9334×10^{0}	0.5916×10^{0}
0.4	20	0.2268×10^{0}	0.1817×10^{2}	0.6042×10^{1}
0.4	50	0.1097×10^{1}	0.3412×10^{2}	0.9126×10^{1}
0.4	100	0.1544×10^{1}	0.1190×10^{3}	0.1520×10^{2}
0.5	10	0.6232×10^{-1}	0.7164×10^{0}	0.5555×10^{0}
0.5	20	0.4839×10^{0}	0.1388×10^{2}	0.4421×10^{1}
0.5	50	0.1257×10^{1}	0.7153×10^{2}	0.8891×10^{1}
0.5	100	0.3031×10^{1}	0.1048×10^{3}	0.2317×10^{2}

APPENDIX III

COMPUTER PROGRAM

A listing of the Fortran program which was used to estimate the five parameters associated with an ellipse is given in this appendix. The program has been broken down into a main program along with several subroutines, each of which has a specific function. Each subroutine is briefly discussed in the following paragraphs.

MAIN Program

The main program performs three functions. The first function is to read all the required input information for the overall program. Secondly, the main program calls the various subroutines in the correct sequence such that the iterations for the estimates of the parameters are correctly performed. Finally, the main program writes out the input information as well as the best estimate for the parameter vector.

The main program that is listed in this appendix is the one which is used in the estimation of the parameters of an ellipse. The main program which is used for the estimation of the parameters of a rectangle is identical to the listed main program except that one statement is added which relates the size of the fitted ellipse to the size of the corresponding rectangle. This scale factor is discussed in Chapter V.

The main program requires the following inputs:

NPAR	the number of parameters which are to be estimated.
NPOINT	the number of data points which lie on the boundary of the unknown pattern.
NTRIAL	the total number of independent local minimizations of the criterion function with respect to the parameter vector. NTRIAL $\geq$ 1.

MSMSQ — the maximum number of times which the LOCMIN subroutine may call the SUMSQR subroutine.

MRAND — the number of independent parameter vectors which are randomly selected by the RANSER subroutine.

NSET — the number of patterns whose parameters are to be estimated.

CE — the initial guess for the unknown parameter vector.

MINPAR — the vector corresponding to the lower limits for the parameters.

MAXPAR — the vector corresponding to the upper limits for the parameters.

SIGMA — the standard deviation of the gaussian noise which is added to the simulated data points generated in the DATA subroutine.

AVE — the average value of the gaussian noise associated with SIGMA.

FSTOP — the upper limit for the absolute value of F. The program is terminated if $|F|$ becomes larger than FSTOP.

EPHI — the lower limit for DPHI. The program is terminated if DPHI becomes smaller than EPHI, where $d_\phi = \frac{\phi(\vec{c}_i) - \phi(\vec{c}_{i+1})}{\phi(\vec{c}_i)}$

EC — the lower limit for DC. The program is terminated if DC becomes smaller than EC, where

$$d_c = \frac{\Delta\vec{c}_i{}^T \Delta\vec{c}_i}{\vec{c}_i{}^T \vec{c}_i}$$

EGRAD — the lower limit for the squared magnitude of the gradient vector. The program is terminated if $|\nabla\phi(c_i)|^2$ becomes smaller than EGRAD.

EBDRY a constant which is used in the GRPREX subroutine to prevent division by zero.

DATA Subroutine

The purpose of the DATA subroutine is to artificially generate the data points which lie on the boundary of either an ellipse or a rectangle. Although the DATA subroutines corresponding to both an ellipse and a rectangle are shown in the listing, only one of them is included in the program when it is actually used. Appendix I gives more details as to how the data points are generated and how simulated noise is added to them.

SUMSQR Subroutine

The SUMSQR subroutine simply evaluates the criterion function for a specific value of the parameter vector. It also has an instability indicator, KX, which is set to one if the absolute value of F exceeds FSTOP.

LOCMIN Subroutine

The LOCMIN subroutine performs a local minimization of the criterion function with respect to the parameters. It does this by calling the next three subroutines. It also checks the various criteria for terminating the program.

REGRES Subroutine

The REGRES subroutine evaluates the criterion function (PHI), the gradient of the criterion function (GRADP), and the Gauss-Newton parameter change vector (BETA) for a specified parameter vector (C) which is supplied by the LOCMIN subroutine. A library subroutine (MINV) is used for matrix inversion.

GRASER Subroutine

The GRASER subroutine is called only when the Newton-Raphson method is used to determine the next value for the parameter vector. The GRASER subroutine finds the optimum binary scale factor by which to multiply $\Delta\vec{c}_i$.

GRPREX Subroutine

The GRPREX subroutine is called only when the full Newton-Raphson step ($\Delta\vec{c}_i = -\frac{\phi(\vec{c}_i)\ \vec{\nabla}\phi(\vec{c}_i)}{|\vec{\nabla}\phi(\vec{c}_i)|^2}$) violates a range constraint. It then projects the gradient onto the constraint surface, after which the GRASER subroutine finds the optimum binary scale factor for this projected gradient. The GRPREX subroutine has an output variable, KEXIT, which when set to one indicates that the parameter vector is on a constraint boundary of the constraint region, R.

RANSER Subroutine

The RANSER subroutine selects a given number (MRAND) of parameter vectors randomly, using a uniform distribution, and determines that parameter vector which yields the smallest value for the criterion function. This parameter vector is then used as the initial guess for another local minimization. The RANSER subroutine uses a library subroutine, RANDU, for its uniform number generator.

A listing of the two library subroutines, RANDU and GAUSS, which generate numbers possessing uniform and normal distributions, respectively, is given at the end of the program listing for the sake of completeness.

```
      SUBROUTINE RANDU (IX, IY, YFL)
      IY = IX  * 65539
      IF (IY) 5, 6, 6
5     IY = IY + 2147483647 + 1
6     YFL = IY
      YFL = YFL * .4656613 E-09
      RETURN
      END


      SUBROUTINE GAUSS (IX, S, AM, V)
      A = 0.0
      DO 50 I = 1, 12
      CALL RANDU (IX, IY, Y)
      IX = IY
50    A = A + Y
      V = (A - 6.0) * S + AM .
      RETURN
      END
```

```
      DIMENSION CO(10),CE(10),C1(10)
      REAL MINPAR(10),MAXPAR(10)
      COMMON/COM1/FSTOP
      COMMON/COM3/NGAUS
      NGAUS=317578125
      K=1
1     CALL SCLOK1
      READ (5,5001) NPAR,NPOINT,NTRIAL,MSMSQ,MRAND,NSET
5001  FORMAT (6I5)
      READ (5,5002) (CO(I),I=1,NPAR),(CE(I),I=1,NPAR),(MINPAR(I),I=1,NPA
     1R),(MAXPAR(I),I=1,NPAR),SIGMA,AVE,FSTOP,EPHI,EC,EGRAD,EBDRY
5002  FORMAT (5F14.8)
      WRITE (6,5003)
5003  FORMAT ('1'   ////////////////////'
     1       NONLINEAR SYSTEM PARAMETER ESTIMATION'///////)
      WRITE (6,5004)
5004  FORMAT ('          NPAR                NPOINT                NTRIAL
     1          MSMSQ                MRAND                NSET'/)
      WRITE (6,5005) NPAR,NPOINT,NTRIAL,MSMSQ,MRAND,NSET
5005  FORMAT (I10,5(10X,I10))
      WRITE (6,5006)
5006  FORMAT (///'            SIGMA              AVE              FSTOP
     1     EPHI             EC             EGRAD             EBDRY'/)
      WRITE (6,5007) SIGMA,AVE,FSTOP,EPHI,EC,EGRAD,EBDRY
5007  FORMAT (T12,F6.3,T28,F5.2,T38,F13.1,T55,F12.10,T69,F12.10,T83,F12.
     110,T100,F12.10)
      WRITE (6,5008)
5008  FORMAT (///'                        I                   CO(I)
     1    CE(I)               MINPAR(I)             MAXPAR(I)'/)
      WRITE (6,5009) (I,CO(I),CE(I),MINPAR(I),MAXPAR(I),I=1,NPAR)
5009  FORMAT (I20,T36,F8.4,T56,F8.4,T76,F8.4,T96,F8.4)
      NSMSQ=0
      CALL DATA (CO,SIGMA,AVE,NPOINT)
      CALL SUMSQR (CE,PHI,KX,NSMSQ,NPOINT,NPAR)
      IF(KX) 3,3,2
2     WRITE (6,5010)
5010  FORMAT (////////////' THE INITIAL PARAMETER ESTIMATES PRODUCE AN UNS
     1TABLE RESPONSE. DESCENT TO A MINIMUM WILL NOT BE CARRIED OUT.')
      GO TO 4
3     CALL LOCMIN (CE,PHI,MINPAR,MAXPAR,EPHI,EC,EGRAD,EBDRY,MSMSQ,NPOINT
     1,NPAR)
      IF(NTRIAL-1) 8,8,4
4     IY=123456789
      DO 7 I=2,NTRIAL
      CALL RANSER (C1,MINPAR,MAXPAR,MRAND,IY,NPOINT,NPAR)
      CALL LOCMIN (C1,PHILOC,MINPAR,MAXPAR,EPHI,EC,EGRAD,EBDRY,MSMSQ,NPO
     1INT,NPAR)
      IF(PHI-PHILOC) 7,7,5
5     PHI=PHILOC
      DO 6 J=1,NPAR
6     CE(J)=C1(J)
7     CONTINUE
8     WRITE (6,5011)
5011  FORMAT ('1'     /////////////////'
     1                FINAL PARAMETER ESTIMATES'/////'
     2 C(1)                  C(2)                  C(3)                  C(4)
     3                C(5)'/)
      WRITE (6,5012) (CE(I),I=1,NPAR)
5012  FORMAT (T15,F13.7,T35,F13.7,T55,F13.7,T75,F13.7,T95,F13.7)
      WRITE (6,5013) PHI
5013  FORMAT (    /////////////////'
     1       SUM-SQUARED ERROR = ',E15.8)
      TIME=RCLOK1(1.)
```

```
      WRITE (6,5014) TIME
5014  FORMAT (//////////'                                    THE TIME REQUIR
     1ED TO EXECUTE THIS PROGRAM WAS',F7.2,' SECONDS.')
9     IF(K-NSET) 10,11,11
10    K=K+1
      GO TO 1
11    CONTINUE
      STOP
      END


C     THIS SUBROUTINE IS USED WHEN DATA POINTS LYING ON THE BOUNDARY OF
C     AN ELLIPSE ARE REQUIRED.
      SUBROUTINE DATA(C,SIGMA,AVE,NPOINT)
      DIMENSION XO(100),YO(100),XD(100),YD(100),C(10)
      COMMON/COM2/X(100),Y(100)
      COMMON/COM3/NGAUS
      DATA IZ$/' '/
      WRITE (6,5001)
5001  FORMAT ('1ENTER DATA SUBROUTINE.'//)
      E1=C(1)
      E2=C(2)
      A=C(3)
      B=C(4)
      TH=C(5)
      RX=SQRT(1./E1)
      XINT=2.*RX
      XO(1)=-RX
      YO(1)=0.
      I=2
1     XO(I)=-RX+I*XINT/NPOINT
      XO(I+1)=XO(I)
      YO(I)=SQRT((1./E2)*(1.-E1*XO(I)**2))
      YO(I+1)=-YO(I)
      I=I+2
      IF (I-NPOINT) 1,2,2
2     XO(I)=RX
      YO(I)=0.
      WRITE (6,5002) C(1),C(2)
5002  FORMAT (1X,' THE FOLLOWING POINTS ARE THE COORDINATES OF AN ELLIPS
     1E HAVING NO DISPLACEMENT FROM THE ORIGIN'/' AND NO ROTATION ABOUT
     2THE ORIGIN. THE EQUATION OF THE ELLIPSE IS GIVEN BY  E1*X*X + E2*Y
     3*Y = 1'/' WHERE E1 =',F10.5,' AND E2 =',F10.5//)
      N$=NPOINT-1
      WRITE (6,5003) (IZ$,XO(I),YO(I),I=1,N$),IZ$,XO(NPOINT),YO(NPOINT)
5003  FORMAT (4(A1,'(',E12.5,',',E12.5,')'))
      DO 3 I=1,NPOINT
      XD(I)=XO(I)*COS(TH)-YO(I)*SIN(TH)+A
      YD(I)=XO(I)*SIN(TH)+YO(I)*COS(TH)+B
3     CONTINUE
      DEGREE=(180./3.1415926536)*TH
      WRITE (6,5004) C(3),C(4),DEGREE
5004  FORMAT(//2X,'THE FOLLOWING POINTS ARE THE COORDINATES OF THE PREVI
     2OUS ELLIPSE WHICH HAS NOW BEEN DISPLACED A =',F10.5/' UNITS IN THE
     3 X-DIRECTION, B =',F10.5,' UNITS IN THE Y-DIRECTION, AND ROTATED B
     3Y THETA =',F10.5,' DEGREES.'//)
      WRITE (6,5005) (IZ$,XD(I),YD(I),I=1,N$),IZ$,XD(NPOINT),YD(NPOINT)
5005  FORMAT (4(A1,'(',E12.5,',',E12.5,')'))
      DO 4 I=1,NPOINT
      CALL GAUSS (NGAUS,SIGMA,AVE,V)
      X(I)=XD(I)+V
      CALL GAUSS (NGAUS,SIGMA,AVE,V)
4     Y(I)=YD(I)+V
```

```
      WRITE (6,5006) AVE,SIGMA
5006  FORMAT(//2X,'THE FOLLOWING POINTS ARE THE PREVIOUS POINTS WITH GAU
     1SSIAN NOISE ADDED. THE MEAN OF THE NOISE IS ',E12.5/' AND THE STAN
     2DARD DEVIATION OF THE NOISE IS ',E12.5//)
      WRITE (6,5007) (IZ$,X(I),Y(I),I=1,N$),IZ$,X(NPOINT),Y(NPOINT)
5007  FORMAT (4(A1,'(',E12.5,',',E12.5,')'))
      WRITE (6,5008)
5008  FORMAT (/' EXIT DATA SUBROUTINE')
      RETURN
      END


C     THIS SUBROUTINE IS USED WHEN DATA POINTS LYING ON THE BOUNDARY OF
C     A RECTANGLE ARE REQUIRED.
      SUBROUTINE DATA(C,SIGMA,AVE,NPOINT)
      DIMENSION C(5),X0(100),Y0(100),XD(100),YD(100),X1(100),Y1(100),X2(
     1100),Y2(100),X3(100),Y3(100),X4(100),Y4(100)
      COMMON/COM2/X(100),Y(100)
      COMMON/COM3/NGAUS
      DATA IZ$/' '/
      WRITE (6,5001)
5001  FORMAT ('1ENTER DATA SUBROUTINE.'//)
      XMAX=1.0/SQRT(C(1))
      YMAX=1.0/SQRT(C(2))
      A=C(3)
      B=C(4)
      TH=C(5)
      CTH=COS(TH)
      STH=SIN(TH)
      YMIN=-YMAX
      XMIN=-XMAX
      NP=NPOINT/4
      XINT=XMAX-XMIN
      YINT=YMAX-YMIN
      N=NP+1
      DO 1 I=1,N
      X1(I)=XMAX
1     Y1(I)=YMIN+(I-1)*YINT/NP
      DO 2 I=2,NP
      X2(I)=XMAX-(I-1)*XINT/NP
2     Y2(I)=YMAX
      DO 3 I=1,N
      X3(I)=XMIN
3     Y3(I)=YMAX-(I-1)*YINT/NP
      DO 4 I=2,NP
      X4(I)=XMIN+(I-1)*XINT/NP
4     Y4(I)=YMIN
      DO 5 I=1,N
      X0(I)=X1(I)
5     Y0(I)=Y1(I)
      N=NP+2
      NN=2*NP
      DO 6 I=N,NN
      X0(I)=X2(I-NP)
6     Y0(I)=Y2(I-NP)
      N=2*NP+1
      NN=3*NP+1
      DO 7 I=N,NN
      X0(I)=X3(I-2*NP)
7     Y0(I)=Y3(I-2*NP)
      N=3*NP+2
      NN=4*NP
      DO 8 I=N,NN
```

```
      XO(I)=X4(I-3*NP)
8     YO(I)=Y4(I-3*NP)
      WRITE (6,5002) XINT,YINT
5002  FORMAT (2X,'THE FOLLOWING POINTS LIE ON A RECTANGLE HAVING NO DISP
     1LACEMENT FROM THE ORIGIN AND NO ROTATION ABOUT THE ORIGIN.'/' THE
     2RECTANGLE IS CENTERED ABOUT THE ORIGIN AND IS',F6.2,'  UNITS IN LE
     3NGTH IN THE X-DIRECTION AND',F6.2,'  UNITS'/' IN LENGTH IN THE Y-D
     4IRECTION.')
      N$=NPOINT-1
      WRITE (6,5003) (IZ$,XO(I),YO(I),I=1,N$),IZ$,XO(NPOINT),YO(NPOINT)
5C03  FORMAT (4(A1,'(',E12.5,',',E12.5,')'))
      DO 9 I=1,NPOINT
      XD(I)=XO(I)*CTH-YO(I)*STH+A
9     YD(I)=XO(I)*STH+YO(I)*CTH+B
      WRITE (6,5004) A,B,TH
5004  FORMAT (//2X,'THE FOLLOWING POINTS ARE THE POINTS ON THE PREVIOUS
     1RECTANGLE WHICH HAS NOW BEEN TRANSLATED',F6.2,'  UNITS IN THE'/' X
     2-DIRECTION,',F6.2,'  UNITS IN THE Y-DIRECTION, AND ROTATED BY',F6.
     32,'  RADIANS.')
      WRITE (6,5005) (IZ$,XD(I),YD(I),I=1,N$),IZ$,XD(NPOINT),YD(NPOINT)
5C05  FORMAT (4(A1,'(',E12.5,',',E12.5,')'))
      DO 10 I=1,NPOINT
      CALL GAUSS(NGAUS,SIGMA,AVE,V)
      X(I)=XD(I)+V
      CALL GAUSS(NGAUS,SIGMA,AVE,V)
10    Y(I)=YD(I)+V
      WRITE (6,5006) AVE,SIGMA
5C06  FORMAT(//2X,'THE FOLLOWING POINTS ARE THE PREVIOUS POINTS WITH GAU
     1SSIAN NOISE ADDED. THE MEAN OF THE NOISE IS ',E12.5/' AND THE STAN
     2DARD DEVIATION OF THE NOISE IS ',E12.5//)
      WRITE (6,5007) (IZ$,X(I),Y(I),I=1,N$),IZ$,X(NPOINT),Y(NPOINT)
5C07  FORMAT (4(A1,'(',E12.5,',',E12.5,')'))
      WRITE (6,5008)
5008  FORMAT (/' EXIT DATA SUBROUTINE')
      RETURN
      END


      SUBROUTINE SUMSQR (C,PHI,KX,NSMSQ,NPOINT,NPAR)
      DIMENSION C(10),F(100)
      COMMON/COM1/FSTOP
      COMMON/COM2/X(100),Y(100)
      NSMSQ=NSMSQ+1
      KX=0
      PHI=0.
      E1=C(1)
      E2=C(2)
      A=C(3)
      B=C(4)
      TH=C(5)
      CTH=COS(TH)
      STH=SIN(TH)
      DO 2 I=1,NPOINT
      F(I)=E1*((X(I)-A)*CTH+(Y(I)-B)*STH)**2+E2*(-(X(I)-A)*STH+(Y(I)-B)*
     1CTH)**2-1.
      IF (ABS(F(I))-FSTOP) 2,1,1
1     KX=1
      WRITE (6,5002) (C(K),K=1,NPAR)
5002  FORMAT (' THE SYSTEM IS UNSTABLE FOR THESE PARAMETER VALUES. THE S
     1UM-SQUARED ERROR WILL NOT BE EVALUATED. THE PARAMETERS ARE'/' C(1)
     2 =',F15.8,' C(2) =',E15.8,' C(3) =',E15.8,' C(4) =',E15.8,' C(5) =
     3',E15.8)
      GO TO 3
```

```
2       PHI = PHI * F(I)*F(I)
3       CONTINUE
        RETURN
        END


        SUBROUTINE LOCMIN (C1,PHI,MINPAR,MAXPAR,EPHI,EC,EGRAD,EBDRY,MSMSQ,
       1NPOINT,NPAR)
        DIMENSION C(10),C1(10),GRADP(10),BETA(10),DELTAC(10)
        REAL MINPAR(10),MAXPAR(10)
        NSMSQ=0
        NS=0
        ND=0
        WRITE (6,5001)
5001    FORMAT ('1ENTER LOCMIN SUBROUTINE.')
1       CALL REGRES (C1,GRADP,BETA,PHI,NPOINT,NPAR)
        PHIINT=PHI
        CSQ=0
        DO 2 I=1,NPAR
2       CSQ=CSQ+C1(I)*C1(I)
        GRADPS=0.
        DO 3 I=1,NPAR
3       GRADPS=GRADPS+GRADP(I)*GRADP(I)
        IF(GRADPS-EGRAD) 4,5,5
4       WRITE (6,5002) GRADPS
5002    FORMAT (/' THE GRADIENT CONDITION IS SATISFIED. THE GRADIENT MAGNI
       1TUDE SQUARED IS GRADS = ',E15.8)
        GO TO 32
5       DO 6 I=1,NPAR
6       C(I)=C1(I)+BETA(I)
        DO 8 I=1,NPAR
        IF(C(I)-MINPAR(I)) 20,20,7
7       IF(C(I)-MAXPAR(I)) 8,20,20
8       CONTINUE
        CALL SUMSQR (C,PHIREG,KX,NSMSQ,NPOINT,NPAR)
        IF(KX) 9,9,21
9       IF(PHIREG-PHI) 10,22,22
10      DO 11 I=1,NPAR
        C1(I)=C(I)
11      DELTAC(I)=BETA(I)
        WRITE (6,5003) PHIREG
5003    FORMAT (/'  THE FULL GAUSS-NEWTON STEP YIELDS A SMALLER VALUE FOR
       1PHI WITHOUT VIOLATING'/' ANY CONSTRAINTS. THE NEW VALUE FOR PHI IS
       2 PHI = ',E15.8)
        WRITE (6,5004)
5004    FORMAT (/'          I           C1(I)'/)
        WRITE (6,5005) (I,C1(I),I=1,NPAR)
5005    FORMAT (I10,E20.8)
        DPHI=(PHI-PHIREG)/PHI
        PHI=PHIREG
        GO TO 13
12      DPHI=(PHI-PHIGRA)/PHI
        PHI=PHIGRA
13      IF(DPHI-EPHI) 14,14,15
14      WRITE (6,5006) DPHI
5006    FORMAT (/' THE NORMALIZED SUM-SQUARED ERROR CHANGE CRITERION IS SA
       1TISFIED. DPHI = ',E15.8)
        GO TO 32
15      DELCSQ=0.
        DO 16 I=1,NPAR
16      DELCSQ=DELCSQ+DELTAC(I)*DELTAC(I)
        DC=DELCSQ/CSQ
        IF(DC-EC) 17,17,18
```

```
17    WRITE (6,5007) DC
5007  FORMAT (/' THE PARAMETER CHANGE ERROR CRITERION IS SATISFIED. DC =
     1 ',E15.8)
      GO TO 32
18    IF(NSMSQ-MSMSQ) 1,19,19
19    WRITE (6,5008)
5008  FORMAT (/' THE LOOP COUNT CRITERION IS SATISFIED.')
      GO TO 32
20    WRITE (6,5009)
5009  FORMAT (/'  THE GAUSS-NEWTON PARAMETER CHANGE VECTOR VIOLATES THE
     1RANGE CONSTRAINTS.'/' A NEWTON-RAPHSON STEP WILL BE TRIED.')
      GO TO 23
21    WRITE (6,5010)
5010  FORMAT (/'  THE GAUSS-NEWTON STEP YIELDED AN UNSTABLE VALUE FOR PH
     1I. THEREFORE,'/' A NEWTON-RAPHSON STEP WILL BE TRIED.')
      GO TO 23
22    WRITE (6,5011)
5011  FORMAT (/'  PHI OBTAINED FROM THE GAUSS-NEWTON STEP WAS GREATER TH
     1AN PHI BEFORE THE STEP WAS TAKEN.'/' THEREFORE, A NEWTON-RAPHSON S
     2TEP WILL BE TRIED.')
23    DO 24 I=1,NPAR
      DELTAC(I)=-(PHI*GRADP(I))/GRADPS
24    C(I)=C1(I)+DELTAC(I)
      WRITE (6,5012)
5012  FORMAT (' THE FULL NEWTON-RAPHSON STEP IS'//'          I          DE
     1LTAC(I)              C(I)'/)
      WRITE (6,5013) (I,DELTAC(I),C(I),I=1,NPAR)
5013  FORMAT (I10,5X,E15.8,5X,E15.8)
      DO 26 I=1,NPAR
      IF(C(I)-MINPAR(I)) 30,25,25
25    IF(C(I)-MAXPAR(I)) 26,26,30
26    CONTINUE
27    CONTINUE
      BINSCL=2.**NS
      DO 28 I=1,NPAR
28    DELTAC(I)=DELTAC(I)/BINSCL
      CALL GRASER (PHIINT,C1,NS,DELTAC,KX,NSMSQ,NPOINT,NPAR)
      PHIGRA=PHIINT
      NT=NS
      NS=ND
      ND=NT
      IF (KX) 29,29,32
29    WRITE (6,5014) ND,NS
5014  FORMAT (/'  THE OPTIMUM DELAY SCALE FACTOR FOUND FROM THE GRASER S
     1UBROUTINE WAS N = ',I2,' AND THE INITIAL DELAY SCALE FACTOR TO BE'
     2/' USED FOR THE NEXT GRASER SUBROUTINE (IF IT IS CALLED AGAIN DURI
     3NG THIS LOCMIN SUBROUTINE) IS N = ',I2)
      WRITE (6,5015) PHIGRA
5015  FORMAT (/' THE GRASER SUBROUTINE FOUND PHI = ',E15.8,'  AND THE PA
     1RAMETERS TO BE'//'          I                C1(I)'/)
      WRITE (6,5016) (I,C1(I),I=1,NPAR)
5016  FORMAT (I10,5X,E20.8)
      GO TO 12
30    WRITE (6,5017)
5017  FORMAT (/'  THE NEWTON-RAPHSON STEP VIOLATES THE RANGE CONSTRAINTS
     1.'/' GRADIENT PROJECTION AND EXTRAPOLATION FOLLOW.')
      CALL GRPREX (C1,DELTAC,MINPAR,MAXPAR,GRADP,KEXIT,EBDRY,NPAR)
      IF(KEXIT) 27,27,31
31    WRITE (6,5018) PHI
5018  FORMAT (/' THE CONSTRAINED MINIMUM IS PHI = ',E15.8,' AND THE PARA
     1METERS ARE'//'          I                C1(I)'/)
      WRITE (6,5019) (I,C1(I),I=1,NPAR)
5019  FORMAT (I10,5X,E15.8)
```

```
32    CONTINUE
      WRITE (6,5020) NSMSQ
5020  FORMAT(//' EXIT LOCMIN SUBROUTINE. NSMSQ = ',I4)
      RETURN
      END


      SUBROUTINE REGRES(C,GRADP,BETA,PHI,NPOINT,NPAR)
      DIMENSION  C(10),GRADP(10),BETA(10),F(100),Z(100,5),S(5,5),L(5),M(
     15)
      COMMON/COM2/X(100),Y(100)
      WRITE (6,5001)
5001  FORMAT (/' ENTER REGRES SUBROUTINE.')
      E1=C(1)
      E2=C(2)
      A=C(3)
      B=C(4)
      TH=C(5)
      CTH=COS(TH)
      STH=SIN(TH)
      DO 1 I=1,NPOINT
      Z(I,1)=((X(I)-A)*CTH+(Y(I)-B)*STH)**2
      Z(I,2)=(-(X(I)-A)*STH+(Y(I)-B)*CTH)**2
      Z(I,3)=2.*(E1*CTH**2+E2*STH**2)*(-X(I)+A) + 2.*(E1-E2)*STH*CTH*(-Y
     1(I)+B)
      Z(I,4)=2.*(E1-E2)*STH*CTH*(-X(I)+A) + 2.*(E1*STH**2+E2*CTH**2)*(-Y
     1(I)+B)
      Z(I,5)=2.*(E1-E2)*((-(X(I)-A)**2+(Y(I)-B)**2)*CTH*STH+(X(I)-A)*(Y(
     1I)-B)*(CTH**2-STH**2))
1     F(I)=E1*((X(I)-A)*CTH+(Y(I)-B)*STH)**2+E2*(-(X(I)-A)*STH+(Y(I)-B)*
     1CTH)**2-1.
      WRITE (6,5002)
5002  FORMAT (//'     I              Z(I,1)              Z(I,2)              Z
     1(I,3)              Z(I,4)              Z(I,5)              F(I)'/)
      WRITE (6,5003)  (I,Z(I,1),Z(I,2),Z(I,3),Z(I,4),Z(I,5),F(I), I=1,
     1NPT2)
5003  FORMAT (I5,5X,6E18.8)
      DO 2 I=1,NPAR
      DO 2 J=1,NPAR
      S(I,J)=0.
      DO 2 K=1,NPOINT
2     S(I,J)=S(I,J)+Z(K,I)*Z(K,J)
      WRITE (6,5004)
5004  FORMAT (//'     I              S(I,1)              S(I,2)              S
     1(I,3)              S(I,4)              S(I,5)'/)
      WRITE (6,5005) (I,S(I,1),S(I,2),S(I,3),S(I,4),S(I,5),I=1,NPAR)
5005  FORMAT (I5,5X,5E18.8)
      CALL MINV(S,NPAR,D,L,M)
      WRITE (6,5006)
5006  FORMAT (//' THE FOLLOWING ARRAY CONTAINS THE ELEMENTS OF S INVERSE
     1. HOWEVER, THEY WILL STILL BE DENOTED BY S.'//'     I              S
     2(I,1)              S(I,2)              S(I,3)              S(I,4)
     3      S(I,5)'/)
      WRITE (6,5005) (I,S(I,1),S(I,2),S(I,3),S(I,4),S(I,5),I=1,NPAR)
      DO 3 I=1,NPAR
      GRADP(I)=0.0
      DO 3 J=1,NPOINT
3     GRADP(I)=GRADP(I)+2.0*Z(J,I)*F(J)
      WRITE (6,5007)
5007  FORMAT (//'                    GRADPHI(1)          GRADPHI(2)          GRA
     1DPHI(3)          GRADPHI(4)          GRADPHI(5)'/)
      WRITE (6,5008) (GRADP(I),I=1,NPAR)
5008  FORMAT (10X,5E18.8)
```

```
      DO 4 I=1,NPAR
      BETA(I)=0.
      DO 4 J=1,NPAR
4     BETA(I)=BETA(I)-(1./2.)*(S(I,J)*GRADP(J))
      PHI=0.
      DO 5 I=1,NPOINT
5     PHI=PHI+F(I)*F(I)
      WRITE (6,5009)
5009  FORMAT (//'     C(1)        C(2)        C(3)        C(4)        C(5)
     1    BETA(1)     BETA(2)     BETA(3)     BETA(4)     BETA(5)      PHI'/)
      WRITE (6,5010) (C(I),I=1,NPAR),(BETA(I),I=1,NPAR),PHI
5010  FORMAT (10(F10.5,1X),E9.2)
      WRITE (6,5011)
5011  FORMAT (/' EXIT REGRES SUBROUTINE.')
      RETURN
      END


      SUBROUTINE GRASER (PHI,C1,N,DELTAC,KX,NSMSQ,NPOINT,NPAR)
      DIMENSION C1(10),C(10),DELTAC(10),DELCMN(10)
      WRITE (6,5001)
5001  FORMAT (//' ENTER GRASER SUBROUTINE.')
      DO 1 I=1,NPAR
1     C(I)=C1(I)+DELTAC(I)
      CALL SUMSQR (C,PHI2,KX,NSMSQ,NPOINT,NPAR)
      IF(KX) 3,3,2
2     WRITE (6,5002)
5002  FORMAT (/' INITIAL DELTAC RESULTS IN AN UNSTABLE SOLUTION.'/' EXIT
     1 GRASER SUBROUTINE.'//)
      GO TO 39
3     DO 4 I=1,NPAR
      DELTAC(I)=DELTAC(I)/2.
4     C(I)=C1(I)+DELTAC(I)
      CALL SUMSQR (C,PHI1,KX,NSMSQ,NPOINT,NPAR)
      IF(KX) 6,6,5
5     WRITE (6,5003)
5003  FORMAT (/' THE STABLE REGION IN PARAMETER SPACE IS NOT CONVEX.'/'
     1EXIT GRASER SUBROUTINE.'//)
      GO TO 39
6     IF(PHI1-PHI2) 9,9,7
7     IF(PHI2-PHI) 16,8,8
8     PHI2=PHI1
      N=N+1
      GO TO 3
9     DO 10 I=1,NPAR
      DELTAC(I)=DELTAC(I)/2.
10    C(I)=C1(I)+DELTAC(I)
      N=N+1
      CALL SUMSQR (C,PHI0,KX,NSMSQ,NPOINT,NPAR)
      IF(KX) 11,11,5
11    IF(PHI1-PHI0) 13,13,12
12    PHI2=PHI1
      PHI1=PHI0
      GO TO 9
13    IF(PHI1-PHI) 14,14,12
14    DO 15 I=1,NPAR
15    DELTAC(I)=2.*DELTAC(I)
      GO TO 32
16    IF(N) 17,17,19
17    WRITE (6,5004)
5004  FORMAT (/' THE STEP FOR N=0 IS LOCALLY MINIMIZING.'/' EXIT GRASER
     1SUBROUTINE.'//)
      DO 18 I=1,NPAR
```

```
18    C1(I)=C1(I)+2.*DELTAC(I)
      PHI=PHI2
      GO TO 39
19    DO 20 I=1,NPAR
      DELTAC(I)=4.*DELTAC(I)
20    C(I)=C1(I)+DELTAC(I)
21    N=N-1
      PHI0=PHI1
      PHI1=PHI2
      CALL SUMSQR (C,PHI2,KX,NSMSQ,NPOINT,NPAR)
      IF(KX) 23,23,22
22    WRITE (6,5005)
5005  FORMAT (/' DECREASING N HAS CAUSED THE SEARCH TO ENTER AN UNSTABLE
     1 REGION.'/' EXIT GRASER SUBROUTINE.'//)
      GO TO 39
23    IF(PHI1-PHI2) 30,24,24
24    IF(N) 25,25,28
25    PHI=PHI2
      WRITE (6,5006)
5006  FORMAT (/' DECREASING N HAS REDUCED N TO ZERO.'/' EXIT GRASER SUBR
     1OUTINE.'//)
26    DO 27 I=1,NPAR
27    C1(I)=C1(I)+DELTAC(I)
      GO TO 39
28    DO 29 I=1,NPAR
      DELTAC(I)=2.*DELTAC(I)
29    C(I)=C1(I)+DELTAC(I)
      GO TO 21
30    DO 31 I=1,NPAR
31    DELTAC(I)=DELTAC(I)/2.
      N=N+1
32    QUADMN=(3./4.)*(PHI2-5.*PHI1+4.*PHI0)/(PHI2-3.*PHI1+2.*PHI0)
      DO 33 I=1,NPAR
      DELCMN(I)=QUADMN*DELTAC(I)
33    C(I)=C1(I)+DELCMN(I)
      CALL SUMSQR (C,PHIMIN,KX,NSMSQ,NPOINT,NPAR)
      IF(KX) 34,34,5
34    IF(PHIMIN-PHI1) 35,37,37
35    DO 36 I=1,NPAR
36    C1(I)=C1(I)+DELCMN(I)
      PHI=PHIMIN
      WRITE (6,5007) PHI
5007  FORMAT (/' THE QUADRATIC FIT FORMULA WAS USED TO COMPUTE DELTAC. T
     1HE MINIMUM VALUE FOUND FOR PHI WAS PHI = ',E15.8,/' EXIT GRASER SU
     2BROUTINE.'//)
      GO TO 39
37    PHI=PHI1
      DO 38 I=1,NPAR
38    C1(I)=C1(I)+DELTAC(I)
      WRITE (6,5008) PHI
5008  FORMAT (/' THE BINARY MINIMUM IS LOWER THAN THE QUADRATIC MINIMUM.
     1 THE MINIMUM VALUE FOUND FOR PHI WAS PHI = ',E15.8,/' EXIT GRASER
     2SUBROUTINE.'//)
39    CONTINUE
      RETURN
      END


      SUBROUTINE GRPREX (C,DELTAC,MINPAR,MAXPAR,GRADP,KEXIT,EBDRY,NPAR)
      DIMENSION C(10),DELTAC(10),GRADP(10),CONBND(10)
      REAL KO,KSTEP,MINPAR(10),MAXPAR(10)
      INTEGER CONBND,CINDEX
      WRITE (6,5001)
```

```
5001  FORMAT(/' ENTER GRPREX SUBROUTINE.')
      CINDEX=0
      KO=0.
      KEXIT=0
      DO 5 I=1,NPAR
      CONBND(I)=0
      IF(C(I)-MINPAR(I)-EBDRY) 1,1,3
1     IF(GRADP(I)) 5,2,2
2     GRADP(I)=0.
      CINDEX=CINDEX+1
      CONBND(I)=1
      GO TO 5
3     IF(C(I)-MAXPAR(I)+EBDRY) 5,4,4
4     IF(GRADP(I)) 2,2,5
5     CONTINUE
      IF(CINDEX-NPAR) 8,6,6
6     WRITE (6,5002)
5002  FORMAT (/' THE SEARCH PROCEDURE HAS ATTAINED A CONSTRAINED MINIMUM
     1.'/' EXIT GRPREX SUBROUTINE.'//)
      KEXIT=1
      DO 7 I=1,NPAR
7     DELTAC(I)=0.
      GO TO 17
8     DO 15 I=1,NPAR
      IF(CONBND(I)) 9,9,15
9     IF(GRADP(I)) 10,15,11
10    KSTEP=(C(I)-MAXPAR(I))/GRADP(I)
      GO TO 12
11    KSTEP=(C(I)-MINPAR(I))/GRADP(I)
12    IF(KO) 14,14,13
13    IF(KSTEP-KO) 14,15,15
14    KO=KSTEP
15    CONTINUE
      DO 16 I=1,NPAR
16    DELTAC(I)=-KO*GRADP(I)
      WRITE (6,5003)
5003  FORMAT (//'                    DELTAC(1)          DELTAC(2)          DEL
     1TAC(3)          DELTAC(4)          DELTAC(5)'/)
      WRITE (6,5004) (DELTAC(I),I=1,NPAR)
5004  FORMAT (10X,5E18.8)
      WRITE (6,5005)
5005  FORMAT (/' GRADIENT PROJECTION AND EXTRAPOLATION TO A BOUNDARY HAS
     1 BEEN ACCOMPLISHED.'/' EXIT GRPREX SUBROUTINE.')
17    CONTINUE
      RETURN
      END


      SUBROUTINE RANSER (C1,MINPAR,MAXPAR,MRAND,IY,NPOINT,NPAR)
      DIMENSION C1(10),C(10),D(10)
      REAL MINPAR(10),MAXPAR(10)
      WRITE (6,5001)
5001  FORMAT ('1ENTER RANSER SUBROUTINE.'/)
      NSMSC=0
      DO 1 I=1,NPAR
1     D(I)=MAXPAR(I)-MINPAR(I)
      IX=IY
2     I=1
3     CONTINUE
      CALL RANDU(IX,IY,YFL)
      IX=IY
      C1(I)=MINPAR(I)+D(I)*YFL
      I=I+1
```

```
      IF (I-NPAR) 3,3,4
4     CALL SUMSQR (C1,PHI,KX,NSMSQ,NPOINT,NPAR)
      IF (KX) 5,5,2
5     WRITE (6,5002) NSMSQ
5002  FORMAT (' RANDOM SEARCHING HAS ESTABLISHED A STARTING VALUE FOR PH
     2I. NSMSQ ='I3/)
      DO 11 I=2,MRAND
      J=1
6     CONTINUE
      CALL RANDU(IX,IY,YFL)
      IX=IY
      C(J)=MINPAR(J)+C(J)*YFL
      J=J+1
      IF (J-NPAR) 6,6,7
7     CALL SUMSQR (C,PHIRAN,KX,NSMSQ,NPOINT,NPAR)
      IF (KX) 8,8,11
8     IF (PHI-PHIRAN) 11,11,9
9     PHI=PHIRAN
      DO 10 J=1,NPAR
10    C1(J)=C(J)
11    CONTINUE
      WRITE (6,5003) PHI
5003  FORMAT (/' THE SMALLEST VALUE FOUND BY THE RANSER SUBROUTINE FOR T
     1HE SUM-SQUARED ERROR IS PHI = 'E15.8//)
      WRITE (6,5004)
5004  FORMAT ('                                                MINIMIZING PARAMETER
     1 VALUES'//'          C(1)                  C(2)                  C(3)
     2                C(4)                  C(5)'/)
      WRITE (6,5005) (C1(I),I=1,NPAR)
5005  FORMAT (5X,F10.5,4(10X,F10.5))
      WRITE (6,5006)
5006  FORMAT (/' EXIT RANSER SUBROUTINE.')
      RETURN
      END
```

REFERENCES

1. Lebedev, D. G. and Lebedev, D. S., "Quantizing Images by Separation and Quantization of Contours", Engrg. Cybernetics, No. 1, pp. 77-81, Jan.-Feb. 1965.

2. Spilerman, S., "A Recognition Logic Designed for Hand Printed Characters", WESCON Convention Record, 1962, pp. 1-7.

3. Zahn, C. T., "A Formal Description for Two-Dimensional Patterns", Proceedings of the International Joint Conference on Artificial Intelligence, ed. by D. Walker and L. Norton, pp. 621-628, May 7-9, 1969.

4. Montanari, U., "A Note on Minimal Length Polygonal Approximation to a Digitized Contour", Communications of the ACM, Vol. 13, No. 1, pp. 41-47, Jan. 1970.

5. Montanari, U., "Continuous Skeletons from Digitized Images", Journal ACM, Vol. 16, No. 4, pp. 534-549, Oct. 1969.

6. Zavalishin, N. V. and Muchnik, I. B., "Linguistic (Structural) Approach to the Pattern Recognition Problems (Review)", Automation and Remote Control, No. 8, pp. 1263-1292, Aug. 1969.

7. Borel, R. J., "A Mathematical Pattern Recognition Technique Based on Contour Shape Properties", M. Sc. thesis, The Ohio State University, Columbus, Ohio (1965).

8. Brill, E. L., "The Development and Evaluation of Three Recognition Criteria for Pattern Recognition", M. Sc. thesis, The Ohio State University, Columbus, Ohio (1967).

9. Graham, D. N., "Image Transmission By Two-Dimensional Contour Coding", Quarterly Progress Report No. 84, Research Laboratory of Electronics, M. I. T., Jan. 15, 1967, pp. 294-301.

10. Freeman, H., "On the Encoding of Arbitrary Geometric Configurations", IRE Transactions on Electronic Computers, Vol. EC-10, No. 2, pp. 260-268, June 1961.

11. Freeman, H., "A Technique for the Classification and Recognition of Geometric Patterns", Proc. 3rd Int'l. Cong. on Cybernetics, Namur, Belgium, pp. 348-369, September 1961.

12. Freeman, H., "Techniques for the Digital Computer Analysis of Chain-Encoded Arbitrary Plane Curves", Proc. National Electronics Conference, Chicago, Illinois, Vol. 17, pp. 421-432, Oct. 1961.

13. Freeman, H., "On the Digital Computer Classification of Geometric Line Patterns", Proc. National Electronics Conference, Chicago, Illinois, Vol. 18, pp. 312-324, Oct. 1962.

14. Freeman, H. and Garder, L., "Apictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition", IEEE Trans. Electronic Computers, EC-13, pp. 118-127, April 1964.

15. Feder, J. and Freeman, H. "Digital Curve Matching Using a Contour Correlation Algorithm", IEEE International Convention Record, Part 3, pp. 69-85, 1966.

16. Freeman, H. and Glass, J. M., "On the Quantization of Line-Drawing Data", IEEE Trans. Systems Science and Cybernetics, SSC-5, pp. 70-79, Jan. 1969.

17. McGhee, R. B., "Some Parameter Optimization Techniques", Chapter 4, 8, Digital Computer User's Handbook, ed. by M. Klerer and G. Korn, McGraw-Hill Book Co., 1967.

NASA-Langley, 1971 — 21 CR-1798